Lecture Notes
in Control and Information Sciences 312

Editors: M. Thoma · M. Morari

# Lecture Notes in Control and Information Sciences

**Edited by M. Thoma and M. Morari**

Further volumes of this series are listed at the end of the book or found on our homepage: springeronline.com

**Vol. 311:** Lamnabhi-Lagarrigue, F.; Loría Perez, J.A.; Panteley, E.V. (Eds.)
Advanced Topics in Control Systems Theory
294 p. 2005 [1-85233-923-3]

**Vol. 310:** Janczak, A.
Identification of Nonlinear Systems
Using Neural Networks and Polynomial Models
323 p. 2005 [3-540-23185-4]

**Vol. 309:** Kumar, V.; Leonard, N.; Morse, A.S. (Eds.)
Cooperative Control
301 p. 2005 [3-540-22861-6]

**Vol. 308:** Tarbouriech, S.; Abdallah, C.T.; Chiasson, J. (Eds.)
Advances in Communication Control Networks
358 p. 2005 [3-540-22819-5]

**Vol. 307:** Kwon, S.J.; Chung, W.K.
Perturbation Compensator based Robust Tracking
Control and State Estimation of Mechanical Systems
158 p. 2004 [3-540-22077-1]

**Vol. 306:** Bien, Z.Z.; Stefanov, D. (Eds.)
Advances in Rehabilitation
472 p. 2004 [3-540-21986-2]

**Vol. 305:** Nebylov, A.
Ensuring Control Accuracy
256 p. 2004 [3-540-21876-9]

**Vol. 304:** Margaris, N.I.
Theory of the Non-linear Analog Phase Locked Loop
303 p. 2004 [3-540-21339-2]

**Vol. 303:** Mahmoud, M.S.
Resilient Control of Uncertain Dynamical Systems
278 p. 2004 [3-540-21351-1]

**Vol. 302:** Filatov, N.M.; Unbehauen, H.
Adaptive Dual Control: Theory and Applications
237 p. 2004 [3-540-21373-2]

**Vol. 301:** de Queiroz, M.; Malisoff, M.; Wolenski, P. (Eds.)
Optimal Control, Stabilization and Nonsmooth Analysis
373 p. 2004 [3-540-21330-9]

**Vol. 300:** Nakamura, M.; Goto, S.; Kyura, N.; Zhang, T.
Mechatronic Servo System Control
Problems in Industries and their Theoretical Solutions
212 p. 2004 [3-540-21096-2]

**Vol. 299:** Tarn, T.-J.; Chen, S.-B.; Zhou, C. (Eds.)
Robotic Welding, Intelligence and Automation
214 p. 2004 [3-540-20804-6]

**Vol. 298:** Choi, Y.; Chung, W.K.
PID Trajectory Tracking Control for Mechanical Systems
127 p. 2004 [3-540-20567-5]

**Vol. 297:** Damm, T.
Rational Matrix Equations in Stochastic Control
219 p. 2004 [3-540-20516-0]

**Vol. 296:** Matsuo, T.; Hasegawa, Y.
Realization Theory of Discrete-Time Dynamical Systems
235 p. 2003 [3-540-40675-1]

**Vol. 295:** Kang, W.; Xiao, M.; Borges, C. (Eds)
New Trends in Nonlinear Dynamics and Control,
and their Applications
365 p. 2003 [3-540-10474-0]

**Vol. 294:** Benvenuti, L.; De Santis, A.; Farina, L. (Eds)
Positive Systems: Theory and Applications (POSTA 2003)
414 p. 2003 [3-540-40342-6]

**Vol. 293:** Chen, G. and Hill, D.J.
Bifurcation Control
320 p. 2003 [3-540-40341-8]

**Vol. 292:** Chen, G. and Yu, X.
Chaos Control
380 p. 2003 [3-540-40405-8]

**Vol. 291:** Xu, J.-X. and Tan, Y.
Linear and Nonlinear Iterative Learning Control
189 p. 2003 [3-540-40173-3]

**Vol. 290:** Borrelli, F.
Constrained Optimal Control
of Linear and Hybrid Systems
237 p. 2003 [3-540-00257-X]

**Vol. 289:** Giarré, L. and Bamieh, B.
Multidisciplinary Research in Control
237 p. 2003 [3-540-00917-5]

**Vol. 288:** Taware, A. and Tao, G.
Control of Sandwich Nonlinear Systems
393 p. 2003 [3-540-44115-8]

**Vol. 287:** Mahmoud, M.M.; Jiang, J. and Zhang, Y.
Active Fault Tolerant Control Systems
239 p. 2003 [3-540-00318-5]

**Vol. 286:** Rantzer, A. and Byrnes C.I. (Eds)
Directions in Mathematical Systems
Theory and Optimization
399 p. 2003 [3-540-00065-8]

**Vol. 285:** Wang, Q.-G.
Decoupling Control
373 p. 2003 [3-540-44128-X]

**Vol. 284:** Johansson, M.
Piecewise Linear Control Systems
216 p. 2003 [3-540-44124-7]

D. Henrion · A. Garulli (Eds.)

# Positive Polynomials in Control

With 21 Figures

Springer

## Editors

Dr. Didier Henrion
LAAS-CNRS
7 Avenue du Colonel Roche
31077 Toulouse
France
and
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
18208 Prague
Czech Republic

Dr. Andrea Garulli
Università di Siena
Dipartimento dell'Informazione
Via Roma, 56
53100 Siena
Italy

ISSN 0170-8643

ISBN 3-540-23948-0 **Springer Berlin Heidelberg New York**

Library of Congress Control Number: 2004117178



**Springer is a part of Springer Science+Business Media**

springeronline.com


Printed in Germany



Typesetting: Data conversion by the authors.
Final processing by PTP-Berlin Protago-TEX-Production GmbH, Germany
Cover-Design: design & production GmbH, Heidelberg
Printed on acid-free paper 62/3141/Yu - 5 4 3 2 1 0

Dedicated to the memory of Jos F. Sturm.

# Preface

Based on the seminal work of Naum Zuselevich Shor (Institute of Cybernetics, Kiev) in the 1980s [1, 2, 3], the theory of positive polynomials lends new theoretical insights into a wide range of control and optimization problems. Positive polynomials can be used to formulate a large number of problems in robust control, non-linear control and non-convex optimization. Only very recently it has been realized that polynomial positivity conditions can be formulated efficiently in terms of Linear Matrix Inequality (LMI) and Semidefinite Programming (SDP) problems. In turn, it is now recognized that LMI and SDP techniques play a fundamental role in convex optimization, see e.g. the plenary talk by Stephen Boyd at the 2002 IEEE Conference on Decision and Control or the successful Workshop on SDP and robust optimization organized in March 2003 by the Institute of Mathematics and its Applications at the University of Minnesota in Minneapolis. For the above reasons, the joint use of positive polynomials and LMI optimization provides an extremely promising approach to difficult control problems.

In the last years, several sessions at major control conferences as well as specialized workshops have been dedicated to these research topics. The invited session *Positive Polynomials in Control* at the 2003 IEEE Conference on Decision and Control, organized by the editors of this volume, has shown that new research directions are quickly emerging, thus pointing out the need for a more detailed overview of the current activity in this research area. This is the main aim of the present book. Another important objective of the book is to collect contributions from several fields (control, optimization, mathematics), in order to show different views and approaches to the topics outlined above.

The book is organized in three parts.

The first part collects a number of articles on applications of positive polynomials and LMI optimization to solve various *control problems*, starting with a contribution by Jarvis-Wloszek, Feeley, Tan, Sun and Packard on the *sum-of-squares (SOS)* decomposition of positive polynomials for nonlinear polynomial systems analysis and design [I.1]. SOS techniques are also

used by Papachristodoulou and Prajna to cope with nonlinear non-polynomial systems, using algebraic reformulation techniques [I.2]. Hol and Scherer in [I.3] describe several results on the use of SOS polynomial matrices to derive bounds on the global optima of non-convex bilinear matrix inequality (BMI) problems, in particular those arising in fixed-order $H_\infty$ design. This latter problem, traditionally deemed as difficult in the control community, is approached differently by Henrion [I.4]: with the help of matrix polynomial positivity conditions, sufficient LMI conditions for scalar fixed-order $H_\infty$ design are obtained. Gram-matrix representation of homogeneous forms, similar to the SOS representation, are used by Chesi, Garulli, Tesi and Vicino in [I.5] to construct less conservative quadratic-in-the-state but polynomial-in-the-parameter Lyapunov functions for assessing robust stability of polytopic linear systems. Finally, positivity conditions for multivariate polynomial matrices are obtained by Bliman [I.6] via the Kalman-Yakubovich-Popov (KYP) lemma, and an application to the design of linear-parameter-varying (LPV) gain-scheduled state-feedback control laws is described.

The second part of the book is more mathematical, and gives an overview of different *algebraic techniques* used to cope with polynomial positivity. Results of semi-algebraic geometry by Hilbert and Pólya led Parrilo [4, 5] to construct converging hierarchies of LMI relaxations for optimization over semi-algebraic sets, based on the *theory of SOS representations* of positive polynomials. Independently, results by Schmüdgen and Putinar were used by Lasserre [6] to construct similar converging LMI hierarchies, with the help of the *theory of moments*. Both Parrilo's and Lasserre's approaches can be viewed as *dual to each other*. The paper by De Klerk, Laurent and Parrilo [II.1] shows equivalence between these two approaches in the special case of minimization of forms on the simplex. In [II.2], Lasserre applies the theory of moments to characterize the set of zeros of triangular sets of polynomial equations. Namely, it is shown that the particular structure of the problem allows for the derivation of a simple LMI formulation. Lasserre's hierarchy of LMI relaxations has proved asymptotic convergence under some constraint qualification assumptions, and in particular if the semi-algebraic feasible set is compact: Powers and Reznick [II.3] investigate what happens with the positivity condition of Schmüdgen-Putinar if this compactness assumption is not satisfied. Finally, in [II.4] Šiljak and Stipanović follow a different approach to ensure polynomial positivity. Based on Bernstein's polynomials, they derive criteria for stability analysis and robust stability analysis of two-indeterminate polynomials.

Finally, the third part of the book is dedicated to *numerical aspects* of positivity of polynomials, and recently developed software tools which can be employed to solve the problems discussed in the book. Parrilo in [III.1] surveys a collection of algebraic results (sparse polynomials and Newton polytopes, ideal structure with equality constraints, structural symmetries) to reduce the size of the LMI formulation of SOS decomposition of positive polynomials. Vandenberghe, Balakrishnan, Wallin, Hansson and Roh [III.2] discuss imple-

mentations of primal-dual interior-point methods for LMI problems derived from the KYP lemma (positivity conditions on one-indeterminate matrix polynomials). It is shown that the overall cost can be reduced to $O(n^4)$, or even $O(n^3)$, as opposed to the $O(n^6)$ of conventional methods, where $n$ is the size of the Lyapunov matrix. In their paper [III.3], Hachez and Nesterov use the theory of conic duality to study in considerable detail optimization problems over positive polynomials with additional interpolation conditions. As a striking result, they show that the complexity of solving the dual LMI formulation is almost independent of the number of interpolation constraints, which has obvious applications in designing more efficient tailored primal-dual interior-point algorithms. The book winds up with descriptions of recent developments in two alternative Matlab software currently available to handle positive multivariate polynomials, using either the SOS decomposition (SOSTOOLS) or the dual moment approach (GloptiPoly). Prajna, Papachristodoulou, Seiler, and Parrilo survey in [III.4] the main features of SOSTOOLS along with its control applications, whereas Henrion and Lasserre in [III.5] describe the global optimality certificate and solution extraction mechanism implemented in GloptiPoly.

We believe that the organization of the book into three parts reflects the current trends in the area, with interplay between control engineers, mathematicians, optimizers and software developers.

October 2004

*Didier Henrion*
*Andrea Garulli*

## References

1. N. Z. Shor (1987). Quadratic optimization problems. Soviet J. Comput. Syst. Sci. 25:1-11.
2. N. Z. Shor (1987). Class of global minimum bounds of polynomial functions. Cybernetics, 23(6):731–734. Russian orig.: Kibernetika, 6:9-11, 1987.
3. N. Z. Shor (1998). Nondifferentiable optimization and polynomial problems. Kluwer, Dordrecht.
4. P. A. Parrilo (2000). Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD Thesis, Calif. Inst. Tech, Pasadena.
5. P. A. Parrilo (2003). Semidefinite programming relaxations for semialgebraic problems. Math. Prog. Ser. B, 96(2):293–320.
6. J. B. Lasserre (2001). Global optimization with polynomials and the problem of moments. SIAM J. Opt. 11(3):796–817.

# Contents

**Part I Control Applications of Polynomial Positivity**

**Control Applications of Sum of Squares Programming**
*Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun and Andrew Packard* ..... 3

**Analysis of Non-polynomial Systems Using the Sum of Squares Decomposition**
*Antonis Papachristodoulou, Stephen Prajna* ..... 23

**A Sum-of-Squares Approach to Fixed-Order $H_\infty$-Synthesis**
*C.W.J. Hol, C.W. Scherer* ..... 45

**LMI Optimization for Fixed-Order $H_\infty$ Controller Design**
*Didier Henrion* ..... 73

**An LMI-Based Technique for Robust Stability Analysis of Linear Systems with Polynomial Parametric Uncertainties**
*Graziano Chesi, Andrea Garulli, Alberto Tesi, Antonio Vicino* ..... 87

**Stabilization of LPV Systems**
*Pierre-Alexandre Bliman* ..... 103

**Part II Algebraic Approaches to Polynomial Positivity**

**On the Equivalence of Algebraic Approaches to the Minimization of Forms on the Simplex**
*Etienne de Klerk, Monique Laurent, Pablo Parrilo* ..... 121

**A Moment Approach to Analyze Zeros of Triangular Polynomial Sets**
*Jean B. Lasserre* ..... 133

**Polynomials Positive on Unbounded Rectangles**
*Victoria Powers, Bruce Reznick* . . . . . 151

**Stability of Interval Two–Variable Polynomials and Quasipolynomials *via* Positivity**
*Dragoslav D. Šiljak, Dušan M. Stipanović* . . . . . 165

**Part III Numerical Aspects of Polynomial Positivity: Structures, Algorithms, Software Tools**

**Exploiting Algebraic Structure in Sum of Squares Programs**
*Pablo A. Parrilo* . . . . . 181

**Interior-Point Algorithms for Semidefinite Programming Problems Derived from the KYP Lemma**
*Lieven Vandenberghe, V. Ragu Balakrishnan, Ragnar Wallin, Anders Hansson, Tae Roh* . . . . . 195

**Optimization Problems over Non-negative Polynomials with Interpolation Constraints**
*Yvan Hachez, Yurii Nesterov* . . . . . 239

**SOSTOOLS and Its Control Applications**
*Stephen Prajna, Antonis Papachristodoulou, Peter Seiler, Pablo A. Parrilo* . . . . . 273

**Detecting Global Optimality and Extracting Solutions in GloptiPoly**
*Didier Henrion, Jean-Bernard Lasserre* . . . . . 293

**Index** . . . . . 311

# Control Applications of Sum of Squares Programming

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard

Department of Mechanical Engineering, University of California, Berkeley
{zachary, rfeeley, weehong, kpsun, pack}@jagger.me.berkeley.edu

We consider nonlinear systems with polynomial vector fields and pose two classes of system theoretic problems that may be solved by sum of squares programming. The first is disturbance analysis using three different norms to bound the reachable set. The second is the synthesis of a polynomial state feedback controller to enlarge the provable region of attraction. We also outline a variant of the state feedback synthesis for handling systems with input saturation. Both classes of problems are demonstrated using two-state nonlinear systems.

## 1 Introduction

Recent developments in sum of squares (SOS) programming [1, 2] have greatly extended the class of problems that can be solved with convex optimization. These results provide a general methodology to find formulations or relaxations, solvable by semidefinite programming, which address seemingly intractable nonconvex problems. Many of the problems that are amenable to SOS programming relate to polynomial optimization or algebraic geometry and reach back to the original work on global lower bounds for polynomials. This work is collected and expanded upon in [3].

First, we define the basic tools needed to state the main theorem, the Positivstellensatz, which leads to the development of our results. We use this methodology to pose two classes of system theoretic problems for nonlinear systems with polynomial vector fields. The first class of problems is disturbance analysis, which we will show three different ways of quantifying the effects of disturbances on polynomial systems:

1. bounding the reachable set subject to unit energy disturbance,
2. bounding the peak bounded disturbance that retains set invariance, and
3. bounding the induced $\mathcal{L}_2 \to \mathcal{L}_2$ gain.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 3–22, 2005.

The second class of problems is expanding a region of attraction with state feedback, and its variant for systems with input saturation. We will illustrate our methods of solving these problems by presenting two proof of concept numerical examples. The two classes of problems presented here is a selection of work done in [4] and [5].

## 2 Preliminaries

We often use the same letter to denote a signal (i.e. a function of time), as well as the possible values that the signal may take on at any time. We hope this abuse of notation will not confuse the reader.

### 2.1 Polynomial Definitions

**Definition 1 (Monomials).** *A **Monomial** $m_\alpha$ in $n$ variables is a function defined as $m_\alpha(x) = x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ for $\alpha \in \mathbb{Z}_+^n$. The degree of a monomial is defined,* $\deg m_\alpha := \sum_{i=1}^n \alpha_i$.

**Definition 2 (Polynomials).** *A **Polynomial** $f$ in $n$ variables is a finite linear combination of monomials,*

$$f := \sum_\alpha c_\alpha m_\alpha = \sum_\alpha c_\alpha x^\alpha$$

*with $c_\alpha \in \mathbb{R}$. Define $\mathcal{R}_n$ to be the set of all polynomials in $n$ variables. The degree of $f$ is defined as* $\deg f := \max_\alpha \deg m_\alpha$ *(provided the associated $c_\alpha$ is non-zero).*

Additionally we define $\Sigma_n$ to be the set of sum of squares (SOS) polynomials in $n$ variables.

$$\Sigma_n := \left\{ p \in \mathcal{R}_n \;\middle|\; p = \sum_{i=1}^t f_i^2 \;,\; f_i \in \mathcal{R}_n,\, i = 1, \ldots, t \right\} .$$

Obviously if $p \in \Sigma_n$, then $p(x) \geq 0 \; \forall x \in \mathbb{R}^n$.

It is interesting to note that there are polynomials that are positive semidefinite (PSD) that are not sum of squares. In general, there are only three combinations of number of variables and degree such that the set of SOS polynomials is equivalent to the set of positive semidefinite ones, namely, $n = 2$; $d = 2$; and $n = 3$ with $d = 4$. This result dates to Hilbert and is related to his 17th problem.

### 2.2 Positivstellensatz

In the section we define concepts to state a central theorem from real algebraic geometry, the Positivstellensatz, which we will hereafter refer to as the P-satz. This is a powerful theorem which generalizes many known results. For example, applying the P-satz, it is possible to derive the $\mathcal{S}$-procedure by carefully picking the free parameters, as will be shown in Sect. 2.4.

**Definition 3.** *Given $\{g_1, \ldots, g_t\} \in \mathcal{R}_n$, the **Multiplicative Monoid** generated by $g_j$'s is the set of all finite products of $g_j$'s, including 1 (i.e. the empty product). It is denoted as $\mathcal{M}(g_1, \ldots, g_t)$. For completeness define $\mathcal{M}(\phi) := 1$.*

An example: $\mathcal{M}(g_1, g_2) = \{g_1^{k_1} g_2^{k_2} \,|\, k_1, k_2 \in \mathbb{Z}_+\}$.

**Definition 4.** *Given $\{f_1, \ldots, f_r\} \in \mathcal{R}_n$, the **Cone** generated by $f_i$'s is*

$$\mathcal{P}(f_1, \ldots, f_r) := \left\{ s_0 + \sum_{i=1}^{l} s_i b_i \;\middle|\; l \in \mathbb{Z}_+, s_i \in \Sigma_n, b_i \in \mathcal{M}(f_1, \ldots, f_r) \right\} . \quad (1)$$

Note that if $s \in \Sigma_n$ and $f \in \mathcal{R}_n$, then $f^2 s \in \Sigma_n$ as well. This allows us to express a cone of $\{f_1, \ldots, f_r\}$ as a sum of $2^r$ terms. An example: $\mathcal{P}(f_1, f_2) = \{s_0 + s_1 f_1 + s_2 f_2 + s_3 f_1 f_2 \,|\, s_0, \ldots, s_3 \in \Sigma_n\}$.

**Definition 5.** *Given $\{h_1, \ldots, h_u\} \in \mathcal{R}_n$, the **Ideal** generated by $h_k$'s is*

$$\mathcal{I}(h_1, \ldots, h_u) := \left\{ \sum_{k=1}^{u} h_k p_k \;\middle|\; p_k \in \mathcal{R}_n \right\} .$$

With these definitions we can state the following theorem taken from [6, Theorem 4.2.2]

**Theorem 1 (Positivstellensatz).** *Given polynomials $\{f_1, \ldots, f_r\}$, $\{g_1, \ldots, g_t\}$, and $\{h_1, \ldots, h_u\}$ in $\mathcal{R}_n$, the following are equivalent:*

1. *The set*
$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} f_1(x) \geq 0, \ldots, f_r(x) \geq 0 \\ g_1(x) \neq 0, \ldots, g_t(x) \neq 0 \\ h_1(x) = 0, \ldots, h_u(x) = 0 \end{array} \right\}$$
*is empty.*
2. *There exist polynomials $f \in \mathcal{P}(f_1, \ldots, f_r)$, $g \in \mathcal{M}(g_1, \ldots, g_t)$, $h \in \mathcal{I}(h_1, \ldots, h_u)$ such that*
$$f + g^2 + h = 0 .$$

When there are only inequality constraints, and they describe a compact region, this theorem can be improved to reduce the number of free parameters [7], and with slightly stronger assumptions [8]. These results have been used to improve bounds on nonconvex polynomial optimization [2] and [9] highlighted a software package to do so.

### 2.3 SOS Programming

Sum of squares polynomials play an important role in the P-satz. Using a "Gram matrix" approach, Choi et al. [10] showed that $p \in \Sigma_n$ iff $\exists\ Q \succeq 0$ such that $p(x) = z^*(x)Qz(x)$, with $z(x)$ a vector of suitable monomials. Powers and Wörmann [11] proposed an algorithm to check if any $Q \succeq 0$ exists for a given $p \in \mathcal{R}_n$. Parrilo [1] showed that their algorithm is an LMI, and proved the following extension.

**Theorem 2 (Parrilo).** *Given a finite set* $\{p_i\}_{i=0}^m \in \mathcal{R}_n$*, the existence of* $\{a_i\}_{i=1}^m \in \mathbb{R}$ *such that*

$$p_0 + \sum_{i=1}^{m} a_i p_i \in \Sigma_n$$

*is an LMI feasibility problem.*

This theorem is useful since it allows one to answer questions like the following SOS programming example.

*Example 1.* Given $p_0, p_1 \in \mathcal{R}_n$, does there exist a $k \in \mathcal{R}_n$, of a given degree, such that

$$p_0 + kp_1 \in \Sigma_n\ . \tag{2}$$

To answer this question, write $k$ as a linear combination of its monomials $\{m_j\}$, $k = \sum_{j=1}^{s} a_j m_j$. Rewrite (2) using this decomposition

$$p_0 + kp_1 = p_0 + \sum_{j=1}^{s} a_j (m_j p_1)$$

which since $(m_j p_1) \in \mathcal{R}_n$ is a feasibility problem that can be checked by Theorem 2.

A software package, SOSTOOLS, [12, 13], exists to aid in solving the LMIs that result from Theorem 2. This package as well as [9] use Sturm's SeDuMi semidefinite programming solver [14].

### 2.4 $\mathcal{S}$-Procedure

What does the $\mathcal{S}$-procedure look like in the P-satz formalism? Given symmetric $n \times n$ matrices $\{A_k\}_{k=0}^m$, the $\mathcal{S}$-procedure states: if there exist nonnegative scalars $\{\lambda_k\}_{k=1}^m$ such that $A_0 - \sum_{k=1}^m \lambda_k A_k \succeq 0$, then

$$\bigcap_{k=1}^{m} \left\{x \in \mathbb{R}^n \mid x^T A_k x \geq 0\right\} \subseteq \left\{x \in \mathbb{R}^n \mid x^T A_0 x \geq 0\right\}\ .$$

Written in P-satz form, the question becomes "is

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} x^T A_1 x \geq 0, \ldots, x^T A_m x \geq 0, \\ -x^T A_0 x \geq 0,\ x^T A_0 x \neq 0 \end{array} \right\}$$

empty?" Certainly, if the $\lambda_k$ exist, define $0 \preceq Q := A_0 - \sum_{k=1}^m \lambda_k A_k$. Further define SOS functions $s_0(x) := x^T Q x, s_{01} := \lambda_1, \ldots, s_{0m} := \lambda_m$. Note that

$$\begin{aligned} f &:= (-x^T A_0 x) s_0 + \textstyle\sum_{k=1}^m (-x^T A_0 x)(x^T A_k x) s_{0k} \\ &\in \mathcal{P}\left(x^T A_1 x, \ldots, x^T A_m x, -x^T A_0 x\right) \end{aligned}$$

and that $g := x^T A_0 x \in \mathcal{M}(x^T A_0 x)$. Substitution yields $f + g^2 = 0$ as desired. We will use this insight to make specific selections in the P-satz formulation of in Sects. 3 and 4. For the special case of $m = 1$, the converse of the $\mathcal{S}$-Procedure is also true [15, Sect. 2.6.3].

Using the tools of SOS programming and the P-satz, we can, after some simplifications, cast some control problems for systems with polynomial vector fields as tractable optimization problems. In the next two sections, we discuss two classes of problems that these techniques are applicable to.

## 3 Disturbance Analysis

In this section, we consider the local effects of external disturbances on polynomial systems. The following types of disturbance analysis are considered:

1. Reachable set bounds under unit energy disturbances
2. Set invariance under peak bounded disturbances
3. Bounding the induced $\mathcal{L}_2 \to \mathcal{L}_2$ gain

### 3.1 Reachable Set Bounds under Unit Energy Disturbances

Given a system of the form

$$\dot{x} = f(x) + g_w(x) w \tag{3}$$

with $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}^{n_w}$, $f \in \mathcal{R}_n^n$, $f(0) = 0$, and $g_w \in \mathcal{R}_n^{n \times n_w}$. We want to compute a bound on the set of points $x(T)$ that are reachable from $x(0) = 0$ under (3), provided the disturbance satisfies $\int_0^T w(t)^* w(t)\, dt \leq 1$, $T \geq 0$.

A similar problem is considered in [16], where real quantifier elimination is used to calculate the exact reachable set for a larger class of dynamical systems. Our approach only considers convex relaxations of the exact problem, and as such requires less computation. A comparison of SOS programming and computational algebra is given in [17] for the case of polynomial minimization.

Following the Lyapunov-like argument in [15, Sect. 6.1.1], if we have a polynomial $V$ such that

$$V(x) > 0 \text{ for all } x \in \mathbb{R}^n \backslash \{0\} \text{ with } V(0) = 0\,, \text{ and} \tag{4}$$

$$\frac{\partial V}{\partial x}(f(x) + g_w(x)w) \leq w^* w \text{ for all } x \in \mathbb{R}^n,\ w \in \mathbb{R}^{n_w}\,, \tag{5}$$

then $\{x|V(x) \leq 1\}$ contains the set of points $x(T)$ that are reachable from $x(0) = 0$ for any $w$ such that $\int_0^T w(t)^* w(t)\, dt \leq 1$, $T \geq 0$. We can see this by integrating the inequality in (5) from 0 to $T$, yielding

$$V(x(T)) - V(x(0)) = \int_0^T w(t)^* w(t)\, dt \leq 1\,.$$

Recalling $V(x(0)) = 0$, $x(T) \in \{x|V(x) \leq 1\}$. Furthermore, $x(\tau) \in \{x|V(x) \leq 1\}$ for all $\tau \in [0, T]$, allowing us to relax the inequality in (5) to

$$\frac{\partial V}{\partial x}(f(x) + g_w(x)w) \leq w^* w \quad \forall x \in \{x \,|\, V(x) \leq 1\},\ \forall w \in \mathbb{R}^{n_w}\,.$$

To bound the reachable set, we require a $V$ satisfying these conditions. Additionally, to achieve a useful bound, the level set $\{x|V(x) \leq 1\}$ should be as small as possible. This is accomplished by requiring $\{x|V(x) \leq 1\}$ to be contained in a variable sized region $P_\beta := \{x \in \mathbb{R}^n | p(x) \leq \beta\}$, for some positive definite $p$, and minimizing $\beta$ under the constraint that we can find a $V$ satisfying (4) and (5). Restricting $V$ to be a polynomial with no constant term, so that $V(0) = 0$, we formulate the problem in the following way, leading to application of the P-satz.

$$\min_{V \in \mathcal{R}_n} \beta$$

such that

$$\{x \in \mathbb{R}^n \mid V(x) \leq 0, l_1(x) \neq 0\} \text{ is empty}\,, \tag{6}$$

$$\{x \in \mathbb{R}^n \mid V(x) \leq 1, p(x) \geq \beta, p(x) \neq \beta\} \text{ is empty}\,, \tag{7}$$

$$\left\{ \begin{array}{l} x \in \mathbb{R}^n\,, \\ w \in \mathbb{R}^{n_w} \end{array} \;\middle|\; \begin{array}{c} V(x) \leq 1, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \geq w^* w, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \neq w^* w \end{array} \right\} \text{ is empty}\,. \tag{8}$$

where $l_1$ is some positive definite and SOS polynomial that replaces $x$ in the non-polynomial constraint $x \neq 0$. The constraints (6) and (8) make $V$ and $\dot{V}$ behave properly, while (7) allows that $\{x|V(x) \leq 1\} \subseteq P_\beta$.

Invoking the P-satz, constraints (6)–(8) are equivalent to the constraints in the following minimization.

$$\min \beta \quad \text{over} \quad \begin{array}{c} V \in \mathcal{R}_n\,, \quad s_1, \ldots, s_6 \in \Sigma_n \\ s_7, \ldots, s_{10} \in \Sigma_{n+n_w}\,, \quad k_1, k_2, k_3 \in \mathbb{Z}_+ \end{array}$$

such that

$$s_1 - V s_2 + l_1^{2k_1} = 0\,, \tag{9}$$

$$\begin{aligned} &s_3 + (1-V)s_4 + (p-\beta)s_5\,, \\ &\quad + (1-V)(p-\beta)s_6 + (p-\beta)^{2k_2} = 0\,, \end{aligned} \tag{10}$$

$$\begin{aligned} &s_7 + \left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^* w\right) s_8 + (1-V)s_9 \\ &\quad + (1-V)\left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^* w\right) s_{10}\,, \\ &\quad + \left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^* w\right)^{2k_3} = 0\,. \end{aligned} \tag{11}$$

Conditions (9)–(11) cannot be directly checked using SOS programming methods. Therefore we specify convenient values for some of the $s_i$'s and $k_j$'s. We also restrict the degree of $V$ and the remaining $s_i$'s. Consequently (9)–(11) become only sufficient for (6)–(8).

Picking any of the $k_j = 0$, can prevent feasibility. Thus we set all of the $k_j$'s equal to the next smallest value, 1. If we pick $s_2 = l_1$ and $s_1 = \hat{s}_1 l_1$, then (9) looks like the form used to show positive definiteness of a Lyapunov function in [18]. Additionally, if we pick $s_7 = s_9 = 0$, and realize that $\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^* w$ is not the zero polynomial, we can write (11) in the form of a "generalized" $\mathcal{S}$-procedure. These choices leave the following problem:

$$\min \beta \quad \text{over } V \in \mathcal{R}_n, \quad s_4, s_5, s_6 \in \Sigma_n, \quad s_{10} \in \Sigma_{n+n_w}$$

such that

$$V - l_1 \in \Sigma_n\,, \tag{12}$$

$$\begin{aligned} -\Big(&(1-V)s_4 + (p-\beta)s_5 \\ &+ (1-V)(p-\beta)s_6 + (p-\beta)^2\Big) \in \Sigma_n\,, \end{aligned} \tag{13}$$

$$-\left((1-V)s_{10} + \left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w) - w^* w\right)\right) \in \Sigma_{n+n_w}\,. \tag{14}$$

where (12) ensures the positive definiteness of $V$, (13) establishes $\{x|V(x) \leq 1\} \subseteq P_\beta$, and (14) constrains $\dot{V} \leq w^* w$ on $\{x|V(x) \leq 1\}$.

Note that some of the decision polynomials enter the constraints in a bilinear form, which SOS programming cannot handle directly. For example, in (13), there are bilinear terms such as $V s_4$ and $V s_6$. Our approach is to hold one set of decision polynomials fixed while optimizing the other set, then switching over. This results in an iterative algorithm whereby at any step, the constraints (12)–(14) can be checked using SOS programming.

Before presenting the algorithm, two issues deserve mention. First, to use SOS programming, we must specify the maximum degree of $V$ and the SOS polynomials $s_i$. To ensure (12)–(14) might be satisfied, the degree of the polynomials must satisfy

$$\begin{gathered}\deg V = \deg l_1 \ , \\ \max\{\deg(Vs_4); \deg(Vps_6)\} \geq \max\{\deg(ps_5); 2\deg p\} \ , \\ \deg s_{10} \geq \max\{\deg f; \deg(g_w w)\} - 1 \ .\end{gathered} \tag{15}$$

These constraints are a consequence of the nature of polynomials; e.g. a SOS polynomial of degree 2 cannot be greater than a SOS polynomial of degree 4 for all $x$.

The second issue is that the algorithm does not reliably find a feasible point $\{V, s_4, s_5, s_6, s_{10}, \beta\}$. Rather it can only improve upon one, by driving $\beta$ smaller. As written, the user must supply an initial $V_0$ that is a component of some feasible point, though the other components can be determined with SDPs. *Given* a $V_0$ satisfying (12), an SDP can determine the existence of $s_4$, $s_5$, and $s_6$ to satisfy (13). Likewise, a separate SDP can determine the existence of $s_{10}$ satisfying (14). Note that a "poor" choice of initial $V_0$ may render (13) and/or (14) unsatisfiable for any choice of $\beta, s_4, s_5, s_6, s_{10}$, although for a different $V_0$, (13) and (14) may be satisfied. Heuristics (based on linearizations) to find suitable initial $V_0$'s are possible. However, once a feasible point $\{V, s_4, s_5, s_6, s_{10}, \beta\}$ is found, the optimization will remain feasible and $\beta$ will be at least monotonically non-increasing with every step of the algorithm. Since we do not have a lower bound on $\beta$, we do not have a formal stopping criteria. Heuristics, such as $\beta$ between each iteration of the algorithm improving by less than a specified tolerance, is used as our stopping criterion.

**Iterative Bounding Procedure**
Setup: Specify the maximum degree that will be considered for both $V$ and the $s_i$'s, observing the constraints in (15). Set $l_1 = \epsilon \sum x_i^m$ for some small $\epsilon > 0$, and $m$ is the maximum degree of $V$. Each step of the iteration, which is indexed by $i$, consists of three substeps, the first two subject to constraints (12)–(14). To begin the iteration, choose a $V_0$, initialize $V^{(i=0)} = V_0$ and the iteration index $i = 1$, and proceed to step 1.

1. **SOS Optimization:**
   Minimize $\beta$ over $s_4, s_5, s_6$, and $s_{10}$, with $V = V^{(i-1)}$ fixed, to obtain $s_4^{(i)}$, $s_6^{(i)}$, and $s_{10}^{(i)}$.
2. **Lyapunov Function Synthesis:**
   Minimize $\beta$ over $s_5$ and $V$, with $s_4 = s_4^{(i)}$, $s_6 = s_6^{(i)}$, and $s_{10} = s_{10}^{(i)}$ fixed, to obtain $V^{(i)}$ and $\beta^{(i)}$.

3. **Stopping Criterion:**
   If $\beta^{(i)} - \beta^{(i-1)}$ is less than a specified tolerance, conclude the iteration, otherwise increment $i$ and return to substep 1.

In (13), $\beta$ is multiplied by polynomials we are searching over. Therefore we minimize $\beta$ in substeps 1 and 2 using a line search.

If we restrict ourselves to linear dynamics, $\dot{x} = Ax + B_w w$, and quadratic Lyapunov functions, $V(x) = x^* P x$, then (12) becomes $P \succ 0$, and with $s_{10} = 0$, (14) becomes

$$\begin{bmatrix} A^*P + PA & PB_w \\ B_w^* P & -I \end{bmatrix} \preceq 0 \, .$$

Thus (12) and (14) generalize the LMI in [15, Sect. 6.1.1].

### 3.2 Set Invariance under Peak Bounded Disturbances

Considering again a polynomial system subject to disturbances as in (3),

$$\dot{x} = f(x) + g_w(x) w \, .$$

We can now look at bounding the maximum peak disturbance value such that a given set remains invariant under these bounded disturbances and the action of the system's dynamics.

Let the peak of a signal $w$ be bounded by

$$\|w\|_\infty := \sup_t |w(t)| \leq \sqrt{\gamma}$$

and define the invariant set as

$$\Omega_1 := \{x \in \mathbb{R}^n | V(x) \leq 1\}$$

for some fixed $V \in \mathcal{R}_n$, positive definite. We know that if $\frac{\partial V}{\partial x}(f(x)+g_w(x)w) \leq 0$ on the boundary of $\Omega_1$ for all $w$ meeting the peak bound, then the flow of the system from any point in $\Omega_1$ cannot ever leave $\Omega_1$, which makes it invariant. In set containment terms we can write this relationship as

$$\{x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} | V(x) = 1\} \cap \{x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} | w^* w \leq \gamma\} \subseteq \left\{ x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} \,\middle|\, \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \leq 0 \right\} \quad (16)$$

which can be rewritten in set emptiness form as

$$\left\{ \begin{array}{l} x \in \mathbb{R}^n, \\ w \in \mathbb{R}^{n_w} \end{array} \,\middle|\, \begin{array}{l} V(x) - 1 = 0, \gamma - w^* w \geq 0, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \geq 0, \\ \frac{\partial V}{\partial x}(f(x) + g_w(x)w) \neq 0 \end{array} \right\} = \phi$$

Employing the P-satz, this becomes

$$\begin{aligned} s_0 &+ s_1(\gamma - w^*w) + s_2\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) \\ &+ s_3(\gamma - w^*w)\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) \\ &+ \left(\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w)\right)^{2k} + q(V-1) = 0 \end{aligned}$$

with $k \in \mathbb{Z}_+$, $q \in \mathcal{R}_{n+n_w}$ and $s_0, s_1, s_2, s_3 \in \Sigma_{n+n_w}$.

Using our standard approach of $k = 1$, we can write the following SOS constraint that guarantees invariance under bounded $w$,

$$\begin{aligned} -\, & s_1(\gamma - w^*w) - s_2\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) \\ & - s_3(\gamma - w^*w)\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) \\ & - \left(\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w)\right)^2 - q(V-1) \in \Sigma_{n+n_w}\,. \end{aligned} \tag{17}$$

Notice that this SOS condition has terms that are not linear in the monomials of $V$, and thus there is no way to use our convex optimization approach to adjust $V$ while checking this condition. Since (17) in linear in $\gamma$ we can search for the maximum peak disturbance for which the set is invariant, by searching over $q$ and the $s_i$'s to maximize $\gamma$ subject to (17). We will need to have the following degree relationship hold to make (17) possibly feasible

$$\begin{aligned} &\max\left\{\deg(s_1) + 2,\ \deg(s_2\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w)),\ \deg(qV)\right\} \\ &\geq \max\left\{\deg(s_3\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w))) + 2,\ 2\deg(\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w)))\right\}\,. \end{aligned}$$

If we set $x(0) = 0$, then the invariant set $\Omega_1$ bounds the system's reachable set under disturbances with peak less than $\gamma$. This bound is similar, but less stringent, than the bound for linear systems given in [15].

The constraint in (17) can result in searching for polynomials with many coefficients. We can reduce the degree of this constraint by setting

$$q = \left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w)\right)^2 \hat{q}$$

and

$$s_i = \left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w)\right)^2 \hat{s}_i$$

for $i = 1, 2, 3$. This allows us to factor out a $\left(\frac{\partial V}{\partial x}(f(x) + g_w(x)w)\right)^2$ term to get the following sufficient condition:

$$\begin{aligned} -\, & \hat{s}_1(\gamma - w^*w) - \hat{s}_2\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) \\ & -\hat{s}_3(\gamma - w^*w)\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w) - 1 - \hat{q}(V-1) \in \Sigma_{n+n_w}\,. \end{aligned} \tag{18}$$

For this simplified constraint (18), the polynomials must satisify this degree relationship:

$$\begin{aligned} \max\left\{\deg(\hat{s}_1) + 2,\ \deg(\hat{s}_2\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w)),\ \deg(\hat{q}V)\right\} \\ \geq \deg\left(\hat{s}_3\tfrac{\partial V}{\partial x}(f(x) + g_w(x)w))\right) + 2\,. \end{aligned} \tag{19}$$

**Effect of $\|w\|_\infty$ on $\|x\|_\infty$**

Using the bounded peak disturbances techniques above to find a bound for the largest disturbance peak value for which $\Omega_1$ is invariant, we can then bound the peak size of the system's state to get a relationship that is similar to the induced $\mathcal{L}_\infty \to \mathcal{L}_\infty$ norm from disturbance to state for this invariant set.

For a given $V$, we solve the optimization to find the largest $\gamma$ such that (17) is feasible. Then we can bound the size of the state by optimizing to find the smallest $\alpha$ such that

$$\Omega_1 = \{x \in \mathbb{R}^n \mid V(x) \leq 1\} \subseteq \{x \in \mathbb{R}^n \mid x^* x \leq \alpha\}$$

This containment constraint is easily solved with a generalized $\mathcal{S}$-procedure following from Sect. 2.4. From this point we know that the following implication holds

$$\forall x(0) \in \Omega_1 \text{ , and } \|w\|_\infty \leq \sqrt{\gamma} \quad \Rightarrow \quad \|x\|_\infty \leq \sqrt{\alpha}\,,$$

which provides our induced norm-like bound.

### 3.3 Bounding the Induced $\mathcal{L}_2 \to \mathcal{L}_2$ Gain

Consider the disturbance driven system with outputs,

$$\begin{aligned} \dot{x} &= f(x) + g_w(x)w \\ y &= h(x) \end{aligned}$$

with $x(t) \in \mathbb{R}^n$, $w(t) \in \mathbb{R}^{n_w}$, $y(t) \in \mathbb{R}^p$, $f \in \mathcal{R}_n^n$, $f(0) = 0$, $g_w \in \mathcal{R}_n^{n \times n_w}$, and $h \in \mathcal{R}_n^p$ with $h(0) = 0$.

For a region, $\Omega_1 = \{x \in \mathbb{R}^n | V(x) \leq 1\}$ as in Sect. 3.2, that is invariant under disturbances with $\|w\|_\infty \leq \sqrt{\gamma}$, we can bound the induced $\mathcal{L}_2 \to \mathcal{L}_2$ gain from $w$ to $y$ on this invariant set by finding a positive definite $H \in \mathcal{R}_n$ and $\beta \geq 0$ such that the following set containment holds

$$\begin{aligned} &\{x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} | w^* w \leq \gamma\} \cap \{x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} | V(x) \leq 1\} \\ &\subseteq \left\{x \in \mathbb{R}^n, w \in \mathbb{R}^{n_w} \,\middle|\, \tfrac{\partial H}{\partial x}(f(x) + g_w(x)w) + h(x)^* h(x) - \beta w^* w \leq 0 \right\} \end{aligned} \quad (20)$$

If we can find a $\beta, H$ pair to make (20) hold, then we can follow the steps from Sect. 3.1 to show that

$$x(0) = 0 \quad \text{and} \quad \|w\|_\infty \leq \sqrt{\gamma} \;\Rightarrow\; \frac{\|y\|_2}{\|w\|_2} \leq \sqrt{\beta}\,.$$

We can search for the tightest bound on the induced norm by employing a generalized $\mathcal{S}$-procedure to satisfy (20) and solving the following optimization

$$\begin{aligned}
&\min_{H\in\mathcal{R}_n} \beta \qquad \text{s.t.}\\
&H - l \in \Sigma_n\,,\\
&-\left(\tfrac{\partial H}{\partial x}(f(x)+g_w(x)w)+h(x)^*h(x)-\beta w^*w\right)\\
&\qquad\qquad -s_1(\gamma - w^*w) - s_2(1-V) \in \Sigma_{n+n_w}
\end{aligned} \tag{21}$$

with $s_1, s_2 \in \Sigma_{n+n_w}$ and $l \in \Sigma_n$, positive definite.

In an effort to make the optimization (21) feasible we will pick the degrees of $s_1$ and $s_2$ so that

$$\begin{aligned}
\deg(s_1) + 2 &\geq \deg\left(\tfrac{\partial H}{\partial x}(f(x)+g_w(x)w)+h^*h\right) \quad \text{and}\\
\deg(s_2 V) &\geq \deg\left(\tfrac{\partial H}{\partial x}(f(x)+g_w(x)w)+h^*h\right)\,.
\end{aligned}$$

### 3.4 Disturbance Analysis Example

Consider the following nonlinear system

$$\begin{aligned}
\dot{x}_1 &= -x_1 + x_2 - x_1x_2^2\\
\dot{x}_2 &= -x_2 - x_1^2x_2 + w\\
y &= [x_1\ x_2]^T
\end{aligned} \tag{22}$$

with $x(t) \in \mathbb{R}^2$ and $w(t) \in \mathbb{R}$. Given $p(x) = 8x_1^2 - 8x_1x_2 + 4x_2^2$, we would like to determine the smallest level set $P_\beta := \{x \in \mathbb{R}^2 \,|\, p(x) \leq \beta\}$ that contains all possible system trajectories for $t \leq T$ starting from $x(0) = 0$ with $\int_0^T w^2\,dt \leq R$, where $R$ is a given constant. Employing the algorithm in Sect. 3.1, we fix $s_5 = 1$ and $s_6 = 0$ to eliminate the need for a line search in each substep. We set the maximum degree of $V$, $s_4$ and $s_{10}$ all to be of degree 4 and initialized the algorithm with $V_0(x) = x_1^2 + x_2^2$. Figure 1 shows the algorithm's progress in reducing $\beta$ versus iteration number as well as the trade off between $R$ and $\beta$. The insert shows the monotonically decreasing behavior of our algorithm for $R = 1$, and after 10 iterations, $\beta$ is reduced to 1.08, which is a large improvement over the first iteration bound of $\beta = 10.79$. For increasing values of disturbance energy $R$, the size of the reachable set increases, which is expected.

Using the Lyapunov function $V$ found in the reachable set analysis for $R = 1$, we can bound the peak disturbance such that the set $\Omega_1 = \{x \in \mathbb{R}^2 \mid V(x) \leq 1\}$ remains invariant. Using the optimization in (18), we get $\|w\|_\infty \leq \sqrt{\gamma} = 0.642$ by choosing the degree of $\hat{s}_1, \hat{s}_2, \hat{s}_3$ and $\hat{p}$ to be 6, 2, 0 and 4 respectively. If we start from $x(0) \in \Omega_1$ and have $\|w\|_\infty \leq 0.642$, then $\|x\|_\infty \leq \sqrt{\alpha} = 0.784$. We can also bound the induced $\mathcal{L}_2 \to \mathcal{L}_2$ disturbance to state gain for this system. The maximum degree of $H, s_1$, and $s_2$ are chosen to be 2, 0 and 2 respectively. Using (21), we get $\frac{\|x\|_2}{\|w\|_2} \leq 1.41$ if we start from $x(0) = 0$, and as long as $\|w\|_\infty \leq 0.642$.

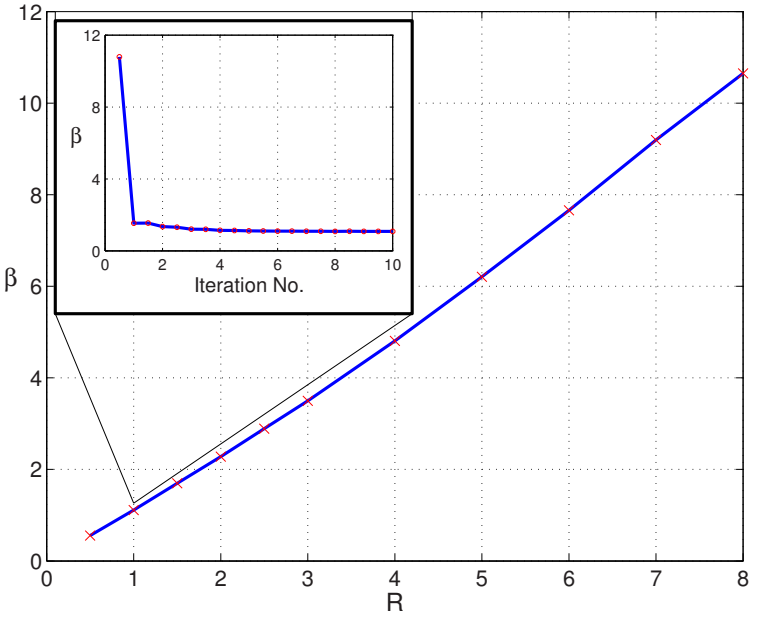



**Fig. 1.** Insert: Algorithm's progress for $R$=1, Main: Trade off between $R$ and $\beta$

## 4 Expanding a Region of Attraction with State Feedback

Given a system of the form

$$\dot{x} = f(x) + g(x)u \tag{23}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, and $f, g$ $n$-vectors of elements of $\mathcal{R}_n$ such that $f(0) = 0$, we would like to synthesize a state feedback controller $u = K(x)$ with $K \in \mathcal{R}_n$ that enlarges the set of points that we can show are attracted to the fixed point at the origin.

We define a variable sized region as $P_\beta := \{x \in \mathbb{R}^n | p(x) \leq \beta\}$, for some given positive definite $p$. We then expand the provable region of attraction by maximizing $\beta$ while requiring that all of the points in $P_\beta$ converge to the origin under the controller $K$. Using a Lyapunov argument, every point in $P_\beta$ will converge asymptotically to the origin if there exists $K, V \in \mathcal{R}_n$ such that the following hold:

$$V(x) > 0 \text{ for all } x \in \mathbb{R}^n \backslash \{0\} \text{ and } V(0) = 0 \,, \tag{24}$$

$$\{x \in \mathbb{R}^n \mid p(x) \leq \beta\} \subseteq \{x \in \mathbb{R}^n \mid V(x) \leq 1\} \,, \tag{25}$$

$$\{x \in \mathbb{R}^n \mid V(x) \leq 1\} \backslash \{0\} \subseteq \left\{ x \in \mathbb{R}^n \;\middle|\; \frac{\partial V}{\partial x}(f(x) + g(x)K(x)) < 0 \right\} \,. \tag{26}$$

These conditions show that $V$ is positive definite, $P_\beta$ is contained in a level set of $V$, and $\frac{dV}{dt}$ is strictly negative on all the points contained in the level set aside from $x = 0$.

The condition that $V(0) = 0$ is satisfied by setting the constant term to zero. Enlarging the region of attraction subject to the preceding requirements can be cast into the following form which is amenable to the P-satz.

$$\max_{K,V \in \mathcal{R}_n} \beta$$

such that

$$\{x \in \mathbb{R}^n \mid V(x) \leq 0,\, l_1(x) \neq 0\} \text{ is empty} \,, \tag{27}$$

$$\{x \in \mathbb{R}^n \mid p(x) \leq \beta,\, V(x) \geq 1,\, V(x) \neq 1\} \text{ is empty} \,, \tag{28}$$

$$\left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} V(x) \leq 1,\; l_2(x) \neq 0, \\ \frac{\partial V}{\partial x}\Big(f(x) + g(x)K(x)\Big) \geq 0 \end{array} \right\} \text{ is empty} \,. \tag{29}$$

where $l_1, l_2$ are fixed positive definite and SOS polynomials which replace the non-polynomial constraints $x \neq 0$ in (24) and (26).

Applying the P-satz, the region maximization problem with constraints (27)–(29) is equivalent to

$$\max \beta \quad \text{over} \quad \begin{array}{c} K, V \in \mathcal{R}_n \quad k_1, k_2, k_3 \in \mathbb{Z}_+ \\ s_1, \ldots, s_{10} \in \Sigma_n \end{array}$$

such that

$$s_1 - V s_2 + l_1^{2k_1} = 0 \,, \tag{30}$$

$$s_3 + (\beta - p)s_4 + (V-1)s_5 + (\beta - p)(V-1)s_6 + (V-1)^{2k_2} = 0 \,, \tag{31}$$

$$s_7 + (1-V)s_8 + (\frac{\partial V}{\partial x}(f + gK))s_9 + (1-V)(\frac{\partial V}{\partial x}(f + gK))s_{10} + l_2^{2k_3} = 0 \,. \tag{32}$$

We cannot check (30)–(32) using SOS programming methods, so we will have to pick values for some of the $s_i$'s and $k_j$'s. We set $k_1 = k_2 = k_3 = 1$ and pick $s_2 = l_1$ and $s_1 = \hat{s}_1 l_1$ to simplify (30). Equation (31) has a $(V-1)^{2k_2}$ term which we can not directly optimize over using SOS programming, so we

cast this constraint as an $\mathcal{S}$-procedure (see Sect. 2.4). This is done by setting $s_3 = s_4 = 0$, $k_2 = 1$, and factoring out a $(V-1)$ term. To simplify (32) we set $s_{10} = 0$ and factor out $l_2$, leaving the sufficient conditions below,

$$\begin{aligned}
&\max \beta \quad \text{over } K, V \in \mathcal{R}_n \quad s_6, s_8, s_9 \in \Sigma_n \\
&\text{such that} \\
&V - l_1 \in \Sigma_n \ , && (33)\\
&-\Big((\beta - p)s_6 + (V-1)\Big) \in \Sigma_n \ , && (34)\\
&-\left((1-V)s_8 + \frac{\partial V}{\partial x}(f+gK)s_9 + l_2\right) \in \Sigma_n \ . && (35)
\end{aligned}$$

Again, the decision polynomials do not enter the constraints linearly, so we employ an iterative algorithm to solve this maximization. A slight modification to (35) is needed because for a given Lyapunov candidate function $V$, searching over $K$ does not affect $\beta$ at all. An intermediate variable, $\alpha$, is introduced to (35) so that we maximize the level set of $\{x \,|\, V(x) \leq \alpha\}$ that is contractively invariant under $K$ and use $\alpha$ to scale $V$ and $l_2$. We will elaborate more in the control design algorithm.

To initialize the algorithm, set $V_0$ to be a control Lyapunov function (CLF) of the linearized system. Since $V_0$ is a CLF, (33) is automatically satisfied and (35) is easily satisfied by scaling $V_0$. Constraint (34) is also satisfied for sufficiently small $\beta$. As such, if we can find a CLF for the linearized system, we would have a feasible starting point for our algorithm. Otherwise, the algorithm might fail on the first iteration.

For reasons highlighted in Sect. 3.1, the maximum degree of $V$, $K$, $l_1$, $l_2$, and the $s_i$'s must satisfy the following constraints:

$$\begin{aligned}
\deg V &= \deg l_1 \ , \\
\deg(ps_6) &\geq \deg V \ , \\
\deg s_8 &\geq \max\{\deg(fs_9); \deg(gKs_9)\} - 1 \ , \\
\deg(Vs_8) &= \deg l_2 \ .
\end{aligned} \tag{36}$$

**Control Design Algorithm**
Setup: Specify the maximum degree that will be considered for both $V$ and the $s_i$'s. Set $l_1 = \epsilon \sum x_i^m$ for some small $\epsilon > 0$, and $m$ is the maximum degree of $V$. Each step of the iteration, indexed by $i$, consists of three substeps, two of which also involve iterations. These inner iterations will be indexed by $j$. To begin the iteration, choose a $V_0$ that is a CLF of the linearized system, and initialize $V^{(i=0)} = V_0$ and $s_9^{(i=0)} = 1$. Also, set $l_2^{(i=0)} = \epsilon \sum x_i^q$, where $q$ is the maximum degree of $(Vs_8)$. Now set the outer iteration index $i = 1$ and proceed to step 1.

1. **Controller Synthesis:**
   Set $V = V^{(i-1)}$, $s_9^{(j=0)} = s_9^{(i-1)}$, and the inner iteration index $j = 1$. In substeps 1a and 1b, solve the following optimization problem:

$$\begin{aligned} &\max \alpha \quad \text{over } K \in \mathcal{R}_n, \quad s_8, s_9 \in \Sigma_n \quad \text{such that} \\ &-\left((\alpha - V)s_8 + \frac{\partial V}{\partial x}(f + gK)s_9 + l_2\right) \in \Sigma_n \ . \end{aligned} \tag{37}$$

   (a) Maximize $\alpha$ over $s_8$, $K$, with $s_9 = s_9^{(j-1)}$ fixed, to obtain $K^{(j)}$.
   (b) Maximize $\alpha$ over $s_8$, $s_9$, with $K = K^{(j)}$ fixed, to obtain $s_9^{(j)}$ and $\alpha^{(j)}$.
   (c) If $\alpha^{(j)} - \alpha^{(j-1)}$ is less than a specified tolerance, set $s_8^{(i)} = s_8^{(j)}$, $s_9^{(i)} = s_9^{(j)}$, $l_2^{(i)} = l_2^{(i-1)}/\alpha^{(j)}$, and $\alpha^{(i)} = \alpha^{(j)}$ and continue to step 2. Otherwise increment $j$ and return to 1a.
2. **Lyapunov Function Synthesis:**
   Set $V^{(j=0)} = V^{(i-1)}/\alpha^{(i)}$ and the inner iteration index $j = 1$. Hold $s_8 = s_8^{(i)}$, $s_9 = s_9^{(i)}$, and $l_2 = l_2^{(i)}$ fixed.
   (a) Maximize $\beta$ over $s_6$, with $V = V^{(j-1)}$ fixed, subject to (34) to obtain $s_6^{(j)}$. i.e.

$$\begin{aligned} &\max \beta \qquad \text{over } s_6 \in \Sigma_n \quad \text{such that} \\ &-\Big((\beta - p)s_6 + (V - 1)\Big) \in \Sigma_n \ . \end{aligned}$$

   (b) Maximize $\beta$ over $V$, with $s_6 = s_6^{(j)}$ fixed, subject to (33)–(35) to obtain $V^{(j)}$ and $\beta^{(j)}$.
   (c) If $\beta^{(j)} - \beta^{(j-1)}$ is less than a specified tolerance, set $V^{(i)} = V^{(j)}$ and $\beta^{(i)} = \beta^{(j)}$ and continue to step 3. Otherwise increment $j$ and return to 2a.
3. **Stopping Criterion:** If $\beta^{(i)} - \beta^{(i-1)}$ is less than a specified tolerance conclude the iterations, otherwise return to step 1.

As in Sect. 3.1, we use a line search to maximize $\alpha$ and $\beta$ in the steps above.

### 4.1 Expanding the Region of Attraction for Systems with Input Saturation

Given a system of the form

$$\dot{x} = f(x) + g(x)\,\mathrm{sat}(u) \tag{38}$$

where

$$\text{sat}(u) := \begin{cases} u \text{ if } |u| \leq 1 \\ 1 \text{ if } u > 1 \\ -1 \text{ if } u < -1 \end{cases}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, and $f, g$ $n$-vectors of elements of $\mathcal{R}_n$ such that $f(0) = 0$, we would like to synthesize a state feedback controller $u = K(x)$ with $K \in \mathcal{R}_n$ to enlarge the set of points which are attracted to the origin.

Again, we define the region to expand as $P_\beta := \{x \in \mathbb{R}^n | p(x) \leq \beta\}$, for some given positive definite $p$. We want to design state feedback controller $K(x)$ to maximize $\beta$ such that the $P_\beta$ is a domain of attraction and $|u| \leq 1$. This is accomplished by appending two conditions to (24)–(26):

$$\{x \in \mathbb{R}^n \mid V(x) \leq 1\} \subseteq \{x \in \mathbb{R}^n \mid K(x) \leq 1\} , \tag{39}$$

$$\{x \in \mathbb{R}^n \mid V(x) \leq 1\} \subseteq \{x \in \mathbb{R}^n \mid K(x) \geq -1\} . \tag{40}$$

These two equations ensure that $|u| = |K(x)| \leq 1$ for all $x$ in the contractively invariant set $\{x \in \mathbb{R}^n | V(x) \leq 1\}$, so the control action will not hit saturation.

Following the procedure in Sect. 4, we will obtain constraints (33)–(35). Additionally due to the saturation, we have

$$\Big((1 - K) - (1 - V)s_{10}\Big) \in \Sigma_n , \tag{41}$$

$$\Big((1 + K) - (1 - V)s_{11}\Big) \in \Sigma_n . \tag{42}$$

The control design algorithm for this problem is similar to that proposed in Sect. 4, with the inclusions of the two additional constraints (41) and (42).

### 4.2 State Feedback Example

Consider the following nonlinear system:

$$\begin{aligned} \dot{x}_1 &= u \\ \dot{x}_2 &= -x_1 + \tfrac{1}{6}x_1^3 - u \end{aligned} \tag{43}$$

with $x(t) \in \mathbb{R}^2$ and $u(t) \in \mathbb{R}$. We are interested in enlarging the domain of attraction described by the level set $P_\beta := \{x \in \mathbb{R}^2 \,|\, p(x) \leq \beta\}$, where $p(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_1x_2 + \frac{1}{12}x_2^2$, through state feedback. Using the algorithm in Sect. 4, we start with randomized $V_0(x)$ that are CLFs of the linearized system. We set the maximum degrees of $V$, $K$, $s_6$, $s_8$ and $s_9$ to 2, 1, 2, 2, and 0 respectively.

Figure 2 shows the progress of $\beta$ with iteration number for 10 random $V_0$. Out of these 10 random $V_0$, the largest $\beta$ achieved is 54.65. Figure 3 shows the resulting domain of attraction for this case. The corresponding controller is $K = -145.94x_1 + 12.2517x_2$ and $V = 0.001(2.3856x_1^2 + 2.108x_1x_2 + 1.17x_2^2)$.

Surprisingly, for higher orders of $V$ and $K$, we have obtained smaller regions of attraction. This is likely due to the nonconvexity of the overall control design algorithm. Although each substep is optimal (i.e., convex), our iterative approach of breaking the algorithm into substeps is not.

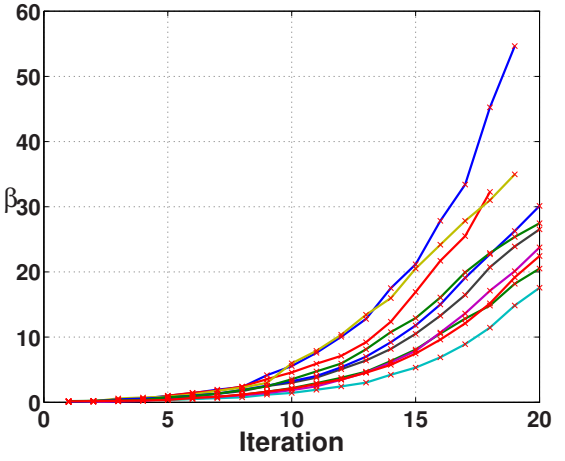



**Fig. 2.** $\beta$ vs. iteration no. for various $V_0$

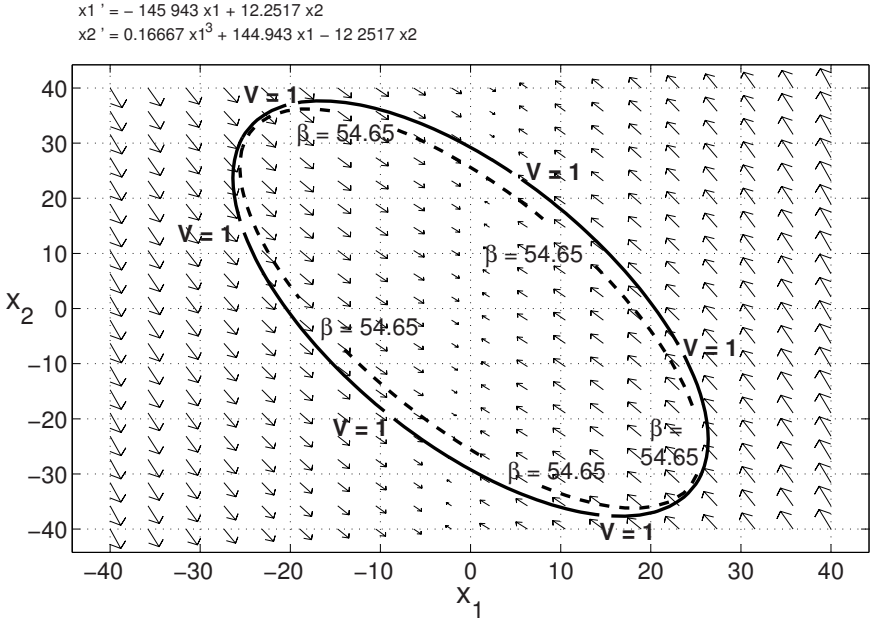



**Fig. 3.** Closed loop system's region of attraction

We can also analyze the disturbance rejection properties of this controller when the disturbances enter the system additively in the control channel, i.e. $g_w(x) = g(x)$. Using the Lyapunov function $V$ found in the state feedback design, we can bound the peak disturbance such that the set $\Omega_1 = \{x \in \mathbb{R}^2 \mid V(x) \leq 1\}$ remains invariant. Using the optimization in (18), we get $\|w\|_\infty \leq \sqrt{\gamma} = 31.62$ by choosing the degree of $\hat{s}_1, \hat{s}_2, \hat{s}_3$ and $\hat{p}$ to be 4, 2, 0 and 4 respectively. If we start with $x(0) \in \Omega_1$ and have $\|w\|_\infty \leq 31.62$, then $\|x\|_\infty \leq \sqrt{\alpha} = 42.22$. We can also bound the induced $\mathcal{L}_2 \to \mathcal{L}_2$ disturbance to state gain for this system by setting $h(x) = [x_1\ x_2]^T$. $H, s_1$, and $s_2$ are all chosen to be of degree 2. Applying (21), if we start from $x(0) = 0$, and as long as $\|w\|_\infty \leq 31.62$, we get $\frac{\|x\|_2}{\|w\|_2} \leq 0.99$.

## 5 Conclusions

Our expansion of existing SOS programming results to two classes of system theoretic questions about nonlinear systems with polynomial vector fields appears promising. The authors believe that there is a multitude of classes of system theoretic questions that can be answered by application of SOS programming. Work in this area is still in its infancy, and the present classes of problems considered is documented in [4].

For the two cases where the decision polynomials do not enter linearly, we resorted to using iterative algorithms. As limited as the two iterative algorithms are, the underlying technique provides opportunities to extend standard LMI analysis of linear systems to more general polynomial vector fields. A drawback of the approach is that implementation of each algorithm requires a feasible starting point. This may be produced by trial and error, or using established nonlinear design techniques.

**Acknowledgements.**
The authors would like to thank the following for providing support for this project: DARPA's Software Enabled Control Program under USAF contract #F33615-99-C-1497, the NSF under contract #CTS-0113985, and DSO National Laboratories-Singapore.

## References

1. P. Parrilo (2003). Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming. 96(2):293–320.
2. J. Lasserre (2000). Global optimization with polynomials and the problem of moments. SIAM Journal of Optimization. 11(3):796–817.

3. N. Shor (1998). Nondiferentiable Optimization and Polynomial Problems. Kluwer Academic Pub, Dordrecht.
4. Z. Jarvis-Wloszek (2003). Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems using Sum-of-Squares Optimization. Ph.D. Dissertation, University of California, Berkeley. Available at `jagger.me.berkeley.edu/~zachary/`.
5. Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard (2003). Some control applications of sum of squares programming. Proc. IEEE Conf. on Decision and Control, 4676–4681.
6. J. Bochnak, M. Coste, and M.-F. Roy (1986). Géométrie algébrique réelle. Springer, Berlin.
7. K. Schmüdgen (1991). The k-moment problem for compact semialgebraic sets. Mathematische Annalen, 289:203–206.
8. M. Putinar (1993). Positive polynomials on compact semialgebraic sets. Indiana University Mathematical Journal, 42:969–984.
9. D. Henrion and J. B. Lasserre (2002). Gloptipoly: Global optimization over polynomials with Matlab and SeDuMi. Proc. IEEE Conf. on Decision and Control, 747–752.
10. M. Choi, T. Lam, and B. Reznick (1995). Sums of squares of real polynomials. Proc. Symposia in Pure Mathematics, 58(2):103–126.
11. V. Powers and T. Wörmann (1998). An algorithm for sums of squares of real polynomials. Journal of pure and applied algebra, 127:99–104.
12. S. Prajna, A. Papachristodoulou, and P. Parrilo (2002). Introducing SOSTOOLS: A general purpose sum of squares programming solver. Proc. IEEE Conf. on Decision and Control, 741–746.
13. S. Prajna, A. Papachristodoulou, and P. A. Parrilo (2002). SOSTOOLS: Sum of squares optimization toolbox for MATLAB.
14. J. Sturm (1999). Using SeDuMi 1.02, a Matlab toolbox for optimizaton over symmetric cones. Optimization Methods and Software, 11–12:625–653. Available at `fewcal.cub.nl/sturm/software/sedumi.html`.
15. S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia.
16. H. Anai and V. Weispfenning (2001). Reach set computations using real quantifier elimination. In M. D. Benedetto and A. Sangiovanni-Vincentelli (Eds.). Hybrid Systems: Comp and Ctrl, LNCS, 2034:63–76, Springer.
17. P. Parrilo and B. Sturmfels (2001). Minimizing polynomial functions. Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science, held at DIMACS, Rutgers University. Available at `control.ee.ethz.ch/~parrilo`.
18. A. Papachristodoulou and S. Prajna (2002). On the construction of Lyapunov functions using the sum of squares decomposition. Proc. IEEE Conf. Decision and Control, 3482–3487.

# Analysis of Non-polynomial Systems Using the Sum of Squares Decomposition

Antonis Papachristodoulou and Stephen Prajna*

Control and Dynamical Systems
California Institute of Technology
Pasadena, CA 91125, USA.
{antonis,prajna}@cds.caltech.edu

Recent advances in semidefinite programming along with use of the sum of squares decomposition to check nonnegativity have paved the way for efficient and algorithmic analysis of systems with polynomial vector fields. In this paper we present a systematic methodology for analyzing the more general class of non-polynomial vector fields, by recasting them into rational vector fields. The sum of squares decomposition techniques can then be applied in conjunction with an extension of the Lyapunov stability theorem to investigate the stability and other properties of the recasted systems, from which properties of the original, non-polynomial systems can be inferred. This will be illustrated by some examples from the mechanical and chemical engineering domains.

## 1 Introduction

The analysis of nonlinear systems has always been a difficult task as the only direct, efficient methodology requires the construction of what is called a *Lyapunov function*. The difficulty lies not only in the 'manual' construction of Lyapunov functions but also in the complexity of testing the non-negativity of the two Lyapunov conditions. Indeed, even if someone was to propose a high order Lyapunov function, it might not at all be possible to verify the two conditions that it needs to satisfy: that it is positive definite in some region around the zero equilibrium and that its derivative along the system's trajectories is non-positive.

Recent advances in the areas of semidefinite programming and the use of the sum of squares decomposition to efficiently check nonnegativity have allowed an algorithmic procedure for systems analysis, something that was

*The authors contributed equally to this work.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 23–43, 2005.

not possible before [10, 9]. Despite these advances, this methodology is restricted to systems described by polynomial vector fields whereas physical systems, the functionality of which is in the focus of many research areas, seldom have polynomial vector fields. For example, it is a common practice to use vector fields with non-rational powers to model enzymatic reactions in biological systems [7]. Also, the model of an aircraft in longitudinal flight contains trigonometric nonlinearities of the angle of attack and pitch angle, but in the same equations one usually captures the coefficients of lift and drag as *polynomial* descriptions of these variables. In this case the stability analysis of the closed loop system using the above methodology becomes difficult, as the same variable appears both in polynomial and non-polynomial terms. The same is true in the case of analysis of chemical processes, where the temperature appears in the energy equation both as a state and also exponentiated in Arrhenius law for the reaction rate.

It has been shown in [14] that any system with non-polynomial nonlinearities can be converted through a simple series of steps to a polynomial system with a larger state dimension, but with a series of equality constraints restricting the states to a manifold of the original state dimension. In some cases the recasting is 'exact', in the sense that the transformed system has a polynomial vector field with the same dimension as the original system — consider for example the case

$$\dot{x}(t) = ce^{-\alpha x(t)}.$$

Setting $p(t) = ce^{-\alpha x(t)}$ we get immediately that

$$\dot{p}(t) = -\alpha p^2(t).$$

In many cases, recasting increases the state dimension but equality constraints that arise from the recasting restrict the system to the original manifold. In particular, the constraints that arise can be either polynomial, or include non-polynomial terms. Consider for example the case of a simple pendulum

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \omega\end{bmatrix} = \begin{bmatrix}\omega\\ -\frac{g}{l}\sin\theta\end{bmatrix}$$

where $g$ is the gravitational constant, $l$ is the length of the pendulum, $\omega$ its angular velocity and $\theta$ the angular deviation of the bead from the vertical. Setting $x_1 = \sin\theta$ and $x_2 = \cos\theta$ one can easily rewrite the above system as

$$\frac{d}{dt}\begin{bmatrix}x_1\\ \omega\\ x_2\end{bmatrix} = \begin{bmatrix}x_2\omega\\ -\frac{g}{l}x_1\\ -x_1\omega\end{bmatrix}$$
$$x_1^2 + x_2^2 = 1$$

where the constraint $x_1^2 + x_2^2 = 1$ is a polynomial equality in $(x_1, x_2)$ that restricts the 3-D recasted system to the 2-D evolution manifold.

However in some other cases, like the case of aircraft dynamics, the case of a reactor process, or the case of enzymatic reactions described by Michaelis-Menten equations these equality constraints are not polynomial, e.g., one would have recastings of the form $x_1 = \sin\alpha$, but then $x_1$ and $\alpha$ also appear in the equations. This is one particularly interesting case that we will explore.

The aim of this chapter is to address stability analysis of non-polynomial systems through a recasting as described above. We extend Lyapunov's stability theorem to handle recasted systems and then use the sum of squares decomposition to construct Lyapunov functions in the new coordinates. When mapped back to the original variables, these Lyapunov functions will contain the original non-polynomial terms.

In Section 2 we review briefly the Lyapunov stability theory for nonlinear systems and how the sum of squares decomposition can be used to construct Lyapunov functions. In Section 3 we present the recasting algorithm and extend the standard Lyapunov theorem to handle the recasted systems. In the examples section, Section 4, we present the analysis of four systems whose vector fields contain radical, trigonometric, irrational power, and exponential terms, which concludes the chapter.

## 2 Background Material

Systems analysis has been in the focus of research for many years. With the development of Lyapunov stability theory, the question of assessing stability of nonlinear systems through the properties of solutions to ordinary differential equations (ODEs) describing the systems was turned into a problem of the existence of what is now known as *Lyapunov functions*. This circumvented the problem of solving the ODE to prove stability. Nonetheless, no explicit algorithm was given on how these functions could be constructed.

The Lyapunov conditions require the construction of a positive definite function that is guaranteed to have non-positive derivative along the trajectories of the system. These conditions are inherently complex, as even testing non-negativity of a polynomial is NP-hard when the polynomial has degree 4 or higher [8]. A sufficient condition for checking non-negativity of a polynomial $p$ is to check whether it admits a sum of squares decomposition [10]; the latter is polynomial-time verifiable — it can be tested by solving a semidefinite programme (SDP).

In this section we give the background material on sum of squares and Lyapunov function theory that will be used in Section 3.

### 2.1 The Sum of Squares Decomposition

We will now give a brief introduction to sum of squares (SOS) polynomials and show how the existence of an SOS decomposition can be verified using semidefinite programming [16]. A more detailed description can be found in [10, 11] and the references therein. We also present briefly an extension of the S-procedure [17, 3] that is used in the main text.

**Definition 1.** *For $x \in \mathbb{R}^n$, a multivariate polynomial $p(x)$ is an SOS if there exist some polynomials $f_i(x)$, $i = 1 \dots M$ such that $p(x) = \sum_{i=1}^{M} f_i^2(x)$.*

An equivalent characterization of SOS polynomials is given in the following proposition.

**Proposition 1.** *A polynomial $p(x)$ of degree $2d$ is an SOS if and only if there exists a positive semidefinite matrix $Q$ and a vector of monomials $Z(x)$ containing all monomials in $x$ of degree $\leq d$ such that $p = Z(x)^T Q Z(x)$.*

The proof of this proposition is based on the eigenvalue decomposition and can be found in [10]. In general, the monomials in $Z(x)$ are not algebraically independent. Expanding $Z(x)^T Q Z(x)$ and equating the coefficients of the resulting monomials to the ones in $p(x)$, we obtain a set of affine relations in the elements of $Q$. Since $p(x)$ being SOS is equivalent to $Q \geq 0$, the problem of finding a $Q$ which proves that $p(x)$ is an SOS can be cast as a semidefinite program (SDP). This was observed by Parrilo in [10].

Note that $p(x)$ being an SOS implies that $p(x) \geq 0$ for all $x \in \mathbb{R}^n$. However, the converse is not always true. Not all nonnegative polynomials can be written as SOS, apart from three special cases: (i) when $n = 2$, (ii) when $\deg(p) = 2$, and (iii) when $n = 3$ and $\deg(p) = 4$. See [13] for more details. Nevertheless, checking nonnegativity of $p(x)$ is an NP-hard problem when the degree of $p(x)$ is at least 4 [8], whereas as argued in the previous paragraph, checking whether $p(x)$ can be written as an SOS is computationally tractable — it can be formulated as an SDP, which has worst-case polynomial time complexity. We will not entail in a discussion on how conservative the relaxation is, but there are several results suggesting that this is not too conservative [13, 11]. Note that as the degree of $p(x)$ and/or its number of variables is increased, the computational complexity for testing whether $p(x)$ is an SOS increases. Nonetheless, the complexity overload is still a polynomial function of these parameters.

There is a close connection between sums of squares and robust control theory through Positivstellensatz, a central theorem in *Real* algebraic geometry [2]. This theorem allows us to formulate a hierarchy of polynomial-time computable stronger conditions [10] for the S-procedure type of analysis [17, 3]. To see how we will be using this result say we want to use the S-procedure to check that the set:

$$\{p(x) \geq 0 \text{ when } p_i(x) \geq 0 \text{ for } i = 1, \ldots, n\}$$

is non-empty. Instead of finding *positive constant* multipliers (the standard S-procedure), we search for *SOS* multipliers $h_i(x)$ so that

$$p(x) - \sum_i h_i(x)p_i(x) \text{ is a SOS.} \tag{1}$$

Since $h_i(x) \geq 0$ and condition (1) is satisfied, for any $x$ such that $p_i(x) \geq 0$ we automatically have $p(x) \geq 0$, so sufficiency follows. This condition is at least as powerful as the standard S-procedure, and many times it is strictly better; it is a special instance of positivstellensatz. By putting an upper bound on the degree of $h_i$ we can get a nested hierarchy of polynomial-time checkable conditions.

Besides this, what is more interesting is the case in which the monomials in the polynomial $p(x)$ have *unknown* coefficients, and we want to search for some values of those coefficients such that $p(x)$ is a sum of squares (and hence nonnegative). Since the unknown coefficients of $p(x)$ are related to the entries of $Q$ via affine constraints, it is evident that the search for the coefficients that make $p(x)$ an SOS can also be formulated as an SDP (these coefficients are themselves decision variables). This observation is crucial in the construction of Lyapunov functions and other S-procedure type multipliers.

Construction of an equivalent SDP for computing SOS decomposition as in Proposition 1 can be quite involved when the degree of the polynomials is high. For this reason, conversion of SOS conditions to the corresponding SDP has been automated in SOSTOOLS [12], a software package developed for this purpose. This software calls SeDuMi [15], an SDP solver to solve the resulting SDP, and converts the solutions back to the solutions of the original SOS programs. These software packages are used for solving all the examples in this chapter.

### 2.2 Lyapunov Stability

Here we concentrate on autonomous nonlinear systems of the form

$$\dot{z} = f(z), \tag{2}$$

where $z \in \mathbb{R}^n$ and for which we assume without loss of generality that $f(0) = 0$, i.e. the origin is an equilibrium of the system. One of the most important properties related to this equilibrium is its stability, and assessing whether stability of the equilibrium holds has been in the center of systems and control research for more than a century. It was not until just before the turn of the 19th century that A. M. Lyapunov formulated sufficient conditions for stability [18] that do not require knowledge of the solution, but are based on the construction of an 'energy-like' function, well known nowadays as a 'Lyapunov function'. Under some technical conditions, the existence of this function was later proved also necessary for asymptotic stability [5].

More precisely, the conditions are stated in the following theorem.

**Theorem 1 (Lyapunov).** *For an open set $\mathcal{D} \subset \mathbb{R}^n$ with $0 \in \mathcal{D}$, suppose there exists a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}$ such that*

$$V(0) = 0, \tag{3}$$
$$V(z) > 0 \quad \forall z \in \mathcal{D} \setminus \{0\}, \tag{4}$$
$$\frac{\partial V}{\partial z}(z) f(z) \leq 0 \quad \forall z \in \mathcal{D}. \tag{5}$$

*Then $z = 0$ is a stable equilibrium of (2).*

It is unfortunate that even with such a powerful theorem, the problem of proving stability of equilibria of nonlinear systems is still difficult; the reason is that there has been no coherent methodology for constructing the Lyapunov function $V(z)$.

In order to simplify the problem at hand, let us assume that $f(z)$ is a polynomial vector field, and that we will be searching for $V(z)$ that is also a polynomial in $z$. Then the two conditions in Theorem 1 become polynomial nonnegativity conditions. To circumvent the difficult task of testing them, we can restrict our attention to cases in which the two conditions admit SOS decompositions. This is the procedure that was originally pursued by Parrilo in his thesis [10]. For $\mathcal{D} = \mathbb{R}^n$, the conditions in Theorem 1 can then be formulated as SOS program stated in the following proposition, and a Lyapunov function that satisfies these conditions can be constructed using semidefinite programming.

**Proposition 2.** *Suppose that for the system (2) there exists a polynomial function $V(z)$ such that*

$$V(0) = 0, \tag{6}$$
$$V(z) - \phi(z) \text{ is SOS}, \tag{7}$$
$$-\frac{\partial V}{\partial z} f(z) \text{ is SOS}, \tag{8}$$

*where $\phi(z) > 0$ for $z \neq 0$. Then the zero equilibrium of (2) is stable.*

*Proof.* Condition (7) enforces $V(z)$ to be positive definite. Since condition (8) implies that $\dot{V}(z)$ is negative semidefinite, it follows that $V(z)$ is a Lyapunov function that proves stability of the origin.

In the above proposition, the function $\phi(z)$ is used to enforce positive definiteness of $V(z)$. If $V(z)$ is a polynomial of degree $2d$, then $\phi(z)$ may be chosen as follows:

$$\phi(z) = \sum_{i=1}^{n} \sum_{j=1}^{d} \epsilon_{ij} z_i^{2j},$$

where the $\epsilon$'s satisfy

$$\sum_{j=1}^{m} \epsilon_{ij} > \gamma \quad \forall\, i = 1, \ldots, n,$$

with $\gamma$ a positive number, and $\epsilon_{ij} \geq 0$ for all $i$ and $j$. In fact, this choice of $\phi(z)$ will force $V(z)$ to be radially unbounded, and hence the stability property holds globally if the conditions in Proposition 2 are met.

## 3 Recasting and Analysis of Recasted Systems

### 3.1 Recasting

In this section we present an algorithm that can be used to convert a non-polynomial system into a rational system. The algorithm is adapted from [14], and it is applicable to a very large class of non-polynomial systems, namely those whose vector field is composed of sums and products of elementary functions, or nested elementary functions of elementary functions. What are meant by elementary functions here are functions with explicit symbolic derivatives such as exponential ($e^x$), logarithm ($\ln x$), power ($x^a$), trigonometric ($\sin x$, $\cos x$, etc.), and hyperbolic functions ($\sinh x$, $\cosh x$, etc.).

Suppose that the original system is given in the form

$$\dot{z}_i = \sum_j \alpha_j \prod_k F_{ijk}(z),$$

where $i = 1, ..., n$; $\alpha_j$'s are real numbers; and $z = (z_1, ..., z_n)$. In the above equation, $F_{ijk}(z)$ are assumed to be elementary functions, or nested elementary functions of elementary functions. For the above system, the recasting algorithm is stated below.

**Algorithm 2 (adopted from [14], with some modifications)**

1. *Let $x_i = z_i$, for $i = 1, ..., n$.*
2. *For each $F_{ijk}(z)$ that is not of the form $F_{ijk}(z) = z_\ell^a$, where $a$ is some integer and $1 \leq \ell \leq n$, introduce a new variable $x_m$. Define $x_m = F_{ijk}(z)$.*
3. *Compute the differential equation describing the time evolution of $x_m$ using the chain rule of differentiation.*
4. *Replace all appearances of such $F_{ijk}(z)$ in the system equations by $x_m$.*
5. *Repeat steps 2–4, until we obtain system equations with rational forms.*

It is best to illustrate the application of the above algorithm by an example.

*Example 1.* Consider the differential equation

$$\dot{z} = \sin(\exp(z) - 1) + 4\ln(z^2 + 1),$$

which we want to recast as a system with rational vector field. We start by defining $x_1 = z$, $x_2 = \sin(\exp(z) - 1)$, and $x_3 = \ln(z^2 + 1)$. By the chain rule of differentiation and replacing the appearances of $z$, $\sin(\exp(z) - 1)$, and $\ln(z^2 + 1)$ in the resulting equations by $x_1$, $x_2$, and $x_3$, we obtain

$$\begin{aligned}
\dot{x}_1 &= x_2 + 4x_3, \\
\dot{x}_2 &= \cos(\exp(z) - 1)\exp(z)\dot{z} \\
&= \cos(\exp(x_1) - 1)\exp(x_1)(x_2 + 4x_3), \\
\dot{x}_3 &= \frac{2}{z^2 + 1} z\dot{z} \\
&= \frac{x_1(x_2 + 4x_3)}{x_1^2 + 1}.
\end{aligned}$$

Notice that the equations for $\dot{x}_1$ and $\dot{x}_3$ are in rational forms. However, the equation for $\dot{x}_2$ is not in a rational form and thus we continue by defining $x_4 = \cos(\exp(x_1) - 1)$ and $x_5 = \exp(x_1)$. Using the chain rule of differentiation again, we obtain

$$\begin{aligned}
\dot{x}_2 &= x_4 x_5 (x_2 + 4x_3), \\
\dot{x}_4 &= -\sin(\exp(x_1) - 1)\exp(x_1)(x_2 + 4x_3) \\
&= -x_2 x_5 (x_2 + 4x_3), \\
\dot{x}_5 &= \exp(x_1)(x_2 + 4x_3) \\
&= x_5 (x_2 + 4x_3).
\end{aligned}$$

At this point, we terminate the recasting process, since the differential equations describing the evolutions of $x_1, ..., x_5$ are already in rational forms.

More examples can be found in Section 4.

### 3.2 Analysis

The recasting process described in the previous subsection generally produces a recasted system whose dimension is higher than the dimension of the original system. To describe the original system faithfully, constraints of the form $x_{n+1} = F(x_1, ..., x_n)$ that are created when new variables are introduced (cf. Algorithm 2) should be taken into account. These constraints define an $n$-dimensional manifold on which the solutions to the original differential equations lie. In general such constraints cannot be converted into polynomial forms, even though sometimes there exist polynomial constraints that are induced by the recasting process. For example:

- Two variables introduced for trigonometric functions such as $x_2 = \sin x_1$, $x_3 = \cos x_1$ are constrained via $x_2^2 + x_3^2 = 1$.
- Introducing a variable to replace a power function such as $x_2 = \sqrt{x_1}$ induces the constraints $x_2^2 - x_1 = 0$, $x_2 \geq 0$.

- Introducing a variable to replace an exponential function such as $x_2 = \exp(x_1)$ induces the constraint $x_2 \geq 0$.

We will shortly discuss how both types of constraints described above can be taken into account in the stability analysis using the sum of squares decomposition technique.

For our purpose, suppose that for a nonpolynomial system

$$\dot{z} = f(z) \tag{9}$$

which has an equilibrium at the origin, the recasted system obtained using the procedure of the previous subsection is written as

$$\dot{\tilde{x}}_1 = f_1(\tilde{x}_1, \tilde{x}_2), \tag{10}$$

$$\dot{\tilde{x}}_2 = f_2(\tilde{x}_1, \tilde{x}_2), \tag{11}$$

where $\tilde{x}_1 = (x_1, ..., x_n) = z$ are the state variables of the original system, $\tilde{x}_2 = (x_{n+1}, ..., x_{n+m})$ are the new variables introduced in the recasting process, and $f_1(\tilde{x}_1, \tilde{x}_2)$, $f_2(\tilde{x}_1, \tilde{x}_2)$ have rational forms.

We denote the constraints that arise directly from the recasting process by

$$\tilde{x}_2 = F(\tilde{x}_1), \tag{12}$$

and those that arise indirectly by

$$G_1(\tilde{x}_1, \tilde{x}_2) = 0, \tag{13}$$

$$G_2(\tilde{x}_1, \tilde{x}_2) \geq 0, \tag{14}$$

where $F$, $G_1$, and $G_2$ are column vectors of functions with appropriate dimensions, and the equalities or inequalities hold entry-wise. The reader should keep in mind that constraints (13)–(14) are actually satisfied only when $\tilde{x}_2 = F(\tilde{x}_1)$ are substituted to (13)–(14). Finally, denote the collective denominator of $f_1(\tilde{x}_1, \tilde{x}_2)$ and $f_2(\tilde{x}_1, \tilde{x}_2)$ by $g(\tilde{x}_1, \tilde{x}_2)$. That is, $g(\tilde{x}_1, \tilde{x}_2)$ should be a polynomial function such that $g(\tilde{x}_1, \tilde{x}_2) f_1(\tilde{x}_1, \tilde{x}_2)$ and $g(\tilde{x}_1, \tilde{x}_2) f_2(\tilde{x}_1, \tilde{x}_2)$ are polynomials. We also assume that $g(\tilde{x}_1, \tilde{x}_2) > 0 \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathcal{D}_1 \times \mathcal{D}_2$, since otherwise the system is not well-posed.

Proving stability of the zero equilibrium of the original system (9) amounts to proving that all trajectories starting close enough to $z = 0$ will remain close to this equilibrium point. This can be accomplished by finding a Lyapunov function $V(z)$ that satisfies the following conditions of Lyapunov's stability theorem, Theorem 1. In terms of the new variables $\tilde{x}_1$ and $\tilde{x}_2$, sufficient conditions that guarantee the existence of a Lyapunov function for the original system are stated in the following proposition.

**Proposition 3.** *Let $\mathcal{D}_1 \subset \mathbb{R}^n$ and $\mathcal{D}_2 \subset \mathbb{R}^m$ be open sets such that $0 \in \mathcal{D}_1$ and $F(\mathcal{D}_1) \subseteq \mathcal{D}_2$. Furthermore, define $\tilde{x}_{2,0} = F(0)$. If there exists a function*

*$\tilde{V} : \mathcal{D}_1 \times \mathcal{D}_2 \to \mathbb{R}$ and column vectors of functions $\lambda_1(\tilde{x}_1, \tilde{x}_2)$, $\lambda_2(\tilde{x}_1, \tilde{x}_2)$, $\sigma_1(\tilde{x}_1, \tilde{x}_2)$, and $\sigma_2(\tilde{x}_1, \tilde{x}_2)$ with appropriate dimensions such that*

$$\tilde{V}(0, \tilde{x}_{2,0}) = 0, \tag{15}$$

$$\begin{aligned}\tilde{V}(\tilde{x}_1, \tilde{x}_2) - \lambda_1^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_1^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) ... \\ \geq \phi(\tilde{x}_1, \tilde{x}_2) \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathcal{D}_1 \times \mathcal{D}_2,\end{aligned} \tag{16}$$

$$\begin{aligned}- g(\tilde{x}_1, \tilde{x}_2) \left( \frac{\partial \tilde{V}}{\partial \tilde{x}_1}(\tilde{x}_1, \tilde{x}_2) f_1(\tilde{x}_1, \tilde{x}_2) + \frac{\partial \tilde{V}}{\partial \tilde{x}_2}(\tilde{x}_1, \tilde{x}_2) f_2(\tilde{x}_1, \tilde{x}_2) \right) ... \\ - \lambda_2^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_2^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) ... \\ \geq 0 \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathcal{D}_1 \times \mathcal{D}_2,\end{aligned} \tag{17}$$

$$\sigma_1(\tilde{x}_1, \tilde{x}_2) \geq 0 \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathbb{R}^{n+m}, \tag{18}$$

$$\sigma_2(\tilde{x}_1, \tilde{x}_2) \geq 0 \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathbb{R}^{n+m}, \tag{19}$$

*for some scalar function $\phi(\tilde{x}_1, \tilde{x}_2)$ with $\phi(\tilde{x}_1, F(\tilde{x}_1)) > 0 \quad \forall \tilde{x}_1 \in \mathcal{D}_1 \setminus \{0\}$, then $z = 0$ is a stable equilibrium of (9).*

*Proof.* Define $V(z) = \tilde{V}(z, F(z))$. From (15) it is straightforward to verify that (3) is satisfied by $V(z)$. Now, (13)–(14), (16), and (18) imply that

$$\begin{aligned}V(\tilde{x}_1, \tilde{x}_2) &\geq \phi(\tilde{x}_1, \tilde{x}_2) + \lambda_1^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) + \sigma_1^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) \\ &\geq \phi(\tilde{x}_1, \tilde{x}_2) \quad \forall (\tilde{x}_1, \tilde{x}_2) \in \mathcal{D}_1 \times \mathcal{D}_2.\end{aligned}$$

Since $\phi(z, F(z)) > 0 \quad \forall z \in \mathcal{D}_1 \setminus \{0\}$ and $F(\mathcal{D}_1) \subseteq \mathcal{D}_2$, it follows that $V(z) > 0 \quad \forall z \in \mathcal{D}_1 \setminus \{0\}$, hence (4) is satisfied.

Finally, by the chain rule of differentiation we have

$$\frac{\partial V}{\partial z}(z) f(z) = \frac{\partial \tilde{V}}{\partial \tilde{x}_1}(z, F(z)) f_1(z, F(z)) + \frac{\partial \tilde{V}}{\partial \tilde{x}_2}(z, F(z)) f_2(z, F(z)),$$

and using the same argument as above in conjunction with (13)–(14), (17), (19), and the fact that $g(\tilde{x}_1, \tilde{x}_2) > 0$, we see that the condition (5) is also satisfied.

Since the conditions (3)–(5) are fulfilled by $V(z)$, we conclude that $V(z)$ is a Lyapunov function for (9) and therefore $z = 0$ is a stable equilibrium of the system.

The above non-negativity conditions can be relaxed to appropriate sum of squares conditions so that they can be algorithmically verified using semidefinite programming, as discussed in Section 2. This will also lead the way to an algorithmic construction of the Lyapunov function $V$. Here we assume that $\mathcal{D}_1 \times \mathcal{D}_2$ is a semialgebraic set described by the following inequalities:

$$\mathcal{D}_1 \times \mathcal{D}_2 = \{(\tilde{x}_1, \tilde{x}_2) \in \mathbb{R}^n \times \mathbb{R}^m : G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2) \geq 0\},$$

where $G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2)$ is a column vector of polynomials and the inequality is satisfied entry-wise. With all this notation, the sum of squares conditions can be stated as follows.

**Proposition 4.** *Let the system (10)–(11) and the functions $F(\tilde{x}_2)$, $G_1(\tilde{x}_1, \tilde{x}_2)$, $G_2(\tilde{x}_1, \tilde{x}_2)$, $G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2)$, and $g(\tilde{x}_1, \tilde{x}_2)$ be given. Define $\tilde{x}_{2,0} = F(0)$. If there exists a polynomial function $\tilde{V}(\tilde{x}_1, \tilde{x}_2)$, column vectors of polynomial functions $\lambda_1(\tilde{x}_1, \tilde{x}_2)$, $\lambda_2(\tilde{x}_1, \tilde{x}_2)$, and column vectors of sum of squares polynomials $\sigma_1(\tilde{x}_1, \tilde{x}_2)$, $\sigma_2(\tilde{x}_1, \tilde{x}_2)$, $\sigma_3(\tilde{x}_1, \tilde{x}_2)$, $\sigma_4(\tilde{x}_1, \tilde{x}_2)$ with appropriate dimensions such that*

$$\tilde{V}(0, \tilde{x}_{2,0}) = 0, \tag{20}$$

$$\begin{aligned}\tilde{V}(\tilde{x}_1, \tilde{x}_2) &- \lambda_1^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_1^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) \ldots \\ &- \sigma_3^T(\tilde{x}_1, \tilde{x}_2) G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2) - \phi(\tilde{x}_1, \tilde{x}_2) \text{ is a sum of squares,}\end{aligned} \tag{21}$$

$$\begin{aligned}- g(\tilde{x}_1, \tilde{x}_2) &\left( \frac{\partial \tilde{V}}{\partial \tilde{x}_1}(\tilde{x}_1, \tilde{x}_2) f_1(\tilde{x}_1, \tilde{x}_2) + \frac{\partial \tilde{V}}{\partial \tilde{x}_2}(\tilde{x}_1, \tilde{x}_2) f_2(\tilde{x}_1, \tilde{x}_2) \right) \ldots \\ &- \lambda_2^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_2^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) \ldots \\ &- \sigma_4^T(\tilde{x}_1, \tilde{x}_2) G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2) \text{ is a sum of squares,}\end{aligned} \tag{22}$$

*for some scalar polynomial function $\phi(\tilde{x}_1, \tilde{x}_2)$ with $\phi(\tilde{x}_1, F(\tilde{x}_1)) > 0 \quad \forall \tilde{x}_1 \in \mathcal{D}_1 \setminus \{0\}$, then $z = 0$ is a stable equilibrium of (9).*

*Proof.* We will show that the above conditions imply the conditions in Proposition 3. Since $\sigma_1(\tilde{x}_1, \tilde{x}_2)$, $\sigma_2(\tilde{x}_1, \tilde{x}_2)$ are sums of squares, (18)–(19) automatically hold. It remains to show that (21)–(22) imply (16)–(17). The condition that (21) is a sum of squares implies that

$$\begin{aligned}\tilde{V}(\tilde{x}_1, \tilde{x}_2) - \lambda_1^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_1^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) \\ \geq \sigma_3^T(\tilde{x}_1, \tilde{x}_2) G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2) + \phi(\tilde{x}_1, \tilde{x}_2)\end{aligned}$$

Now, for $(\tilde{x}_1, \tilde{x}_2) \in \mathcal{D}_1 \times \mathcal{D}_2$, we have $\sigma_3(\tilde{x}_1, \tilde{x}_2) G_{\mathcal{D}}(\tilde{x}_1, \tilde{x}_2) \geq 0$ and therefore it follows that for any such $(\tilde{x}_1, \tilde{x}_2)$,

$$\tilde{V}(\tilde{x}_1, \tilde{x}_2) - \lambda_1^T(\tilde{x}_1, \tilde{x}_2) G_1(\tilde{x}_1, \tilde{x}_2) - \sigma_1^T(\tilde{x}_1, \tilde{x}_2) G_2(\tilde{x}_1, \tilde{x}_2) \geq \phi(\tilde{x}_1, \tilde{x}_2),$$

which is (16). Using the same argument as above, it is straightforward to show that (17) is also fulfilled.

## 4 Examples

Here we present four examples. In the first example the vector field of the system includes a radical term. Such terms appear frequently when considering

systems with saturation nonlinearities. The second example is a mechanical system, whose description contains trigonometric terms that appear when considering the system in cylindric coordinates. The third example shows how one can analyze non-polynomial vector fields with irrational powers, which for example appear in models of biological systems. In the last example we analyze a system that appears frequently in chemical engineering, that of a diabatic Continuous Stirred Tank Reactor (CSTR) with a single first-order exothermic irreversible reaction $A \rightarrow B$.

### 4.1 Example 4.1: System with Saturation Nonlinearity

For a general system, finding a global Lyapunov function is difficult, as one of polynomial form in the variables considered might not exist. However, if a term is expected to appear in a Lyapunov function, the search can be directed to include that term in the Lyapunov function expression sought, by changing variables and recasting the system in an equivalent one in terms of that variable, making use of inequality and equality constraints.

Consider, for example, the system

$$\dot{x}_1 = x_2, \tag{23}$$
$$\dot{x}_2 = -\varphi(x_1 + x_2), \tag{24}$$

where the function $\varphi$ is a saturation function of the following form:

$$\varphi(\sigma) = \frac{\sigma}{\sqrt{1+\sigma^2}}.$$

This function has only one equilibrium point, namely the origin. Notice that for the above memoryless nonlinearity,

$$\Phi(\sigma) = \int_0^\sigma \varphi(\tau) d\tau = \sqrt{1+\sigma^2} - 1 \tag{25}$$

is a positive definite function. In fact, we expect terms of the above form to appear in the Lyapunov function.

First rewrite the above system in a polynomial form, as it has non-polynomial terms. For this purpose, introduce the following auxiliary variables:

$$\begin{aligned} u_1 &= \sqrt{1+(x_1+x_2)^2}, \\ u_2 &= 1/u_1, \\ u_3 &= \sqrt{1+x_1^2}, \\ u_4 &= 1/u_3. \end{aligned}$$

Then the equations of motion for the above system become

$$\begin{aligned}
\dot{x}_1 &= x_2,\\
\dot{x}_2 &= -(x_1+x_2)u_2,\\
\dot{u}_1 &= (x_1+x_2)(x_2-x_1u_2-x_2u_2)u_2,\\
\dot{u}_2 &= -(x_1+x_2)(x_2-x_1u_2-x_2u_2)u_2^3,\\
\dot{u}_3 &= x_1x_2u_4,\\
\dot{u}_4 &= -x_1x_2u_4^3.
\end{aligned}$$

In addition, we have a number of equality and inequality constraints

$$\begin{aligned}
u_1^2 &= 1+(x_1+x_2)^2,\\
u_1u_2 &= 1,\\
u_3^2 &= 1+x_1^2,\\
u_3u_4 &= 1,\\
u_i &\geq 0, \quad \text{for } i=1,2,3,4.
\end{aligned}$$

Now the system is in the form (10)–(11) with $\tilde{x}_1 = (x_1, x_2)$ and $\tilde{x}_2 = (u_1, u_2, u_3, u_4)$. The above constraints correspond to Equation (12). The new representation allows us to use SOS decomposition to compute a Lyapunov function for this problem, using Proposition 4. Terms that are nonpolynomial in $x_1$ and $x_2$, such as $u_3$, can be included in the search, and indeed they should be, as they have the same form as the positive definite function (25). Thus, for example, we may search for a Lyapunov function of the following form:

$$V = a_1 + a_2u_3 + a_3x_1^2 + a_4x_1x_2 + a_5x_2^2,$$

where the $a_i$'s are the unknowns, with $a_1 + a_2 = 0$, so that $V$ is equal to zero at $(x_1, x_2) = (0, 0)$. To guarantee positive definiteness, we require $V$ to satisfy

$$(V - \epsilon_1(u_3-1) - \epsilon_2x_1^2 - \epsilon_3x_2^2) \quad \text{is a sum of squares,}$$

with $\epsilon_1$, $\epsilon_2$, $\epsilon_3$ being non-negative decision variables that satisfy, for example,

$$\begin{aligned}
\epsilon_1+\epsilon_2 &\geq 0.1,\\
\epsilon_3 &\geq 0.1.
\end{aligned}$$

Using this method, a Lyapunov function has been constructed for the system:

$$\begin{aligned}
V &= -3.9364 + 3.9364u_3 + 0.0063889x_1^2 + 0.010088x_1x_2 + 2.0256x_2^2\\
&= -3.9364 + 3.9364\sqrt{1+x_1^2} + 0.0063889x_1^2 + 0.010088x_1x_2 + 2.0256x_2^2.
\end{aligned}$$

The level curves of this Lyapunov function are shown in Figure 1.

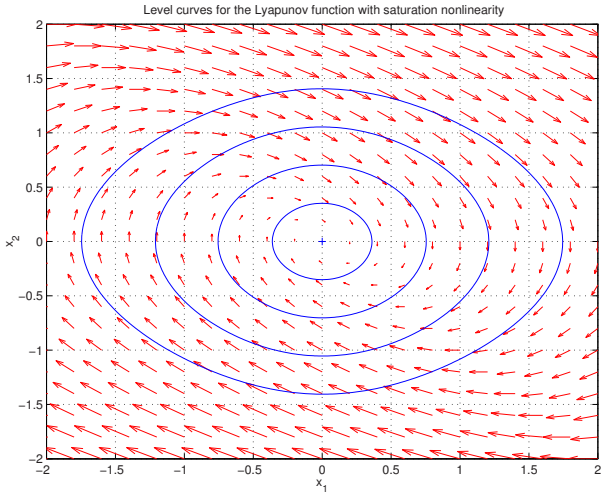



**Fig. 1.** Global Lyapunov function for the system with saturation nonlinearity. Arrows show vector field, solid lines show level curves of the Lyapunov function.

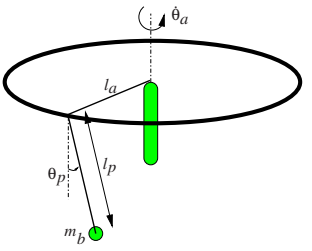



**Fig. 2.** The whirling pendulum

### 4.2 Example 4.2: Whirling Pendulum

Consider the whirling pendulum [4] shown in Figure 2. It is a pendulum of length $l_p$ whose suspension end is attached to a rigid arm of length $l_a$, with a mass $m_b$ attached to its free end. The arm rotates with angular velocity $\dot{\theta}_a$. The pendulum can oscillate with angular velocity $\dot{\theta}_p$ in a plane normal to the arm, making an angle $\theta_p$ with the vertical in the instantaneous plane of motion. We will ignore frictional effects and assume that all links are slender so that their moment of inertia can be neglected.

Using $x_1 = \theta_p$ and $x_2 = \dot{\theta}_p$ as state variables, we obtain the following state equations for the system:

$$\dot{x}_1 = x_2, \tag{26}$$

$$\dot{x}_2 = \dot{\theta}_a^2 \sin x_1 \cos x_1 - \frac{g}{l_p} \sin x_1. \tag{27}$$

The number and stability properties of equilibria in this system depend on the value of $\dot{\theta}_a$. When the condition

$$\dot{\theta}_a^2 < g/l_p \tag{28}$$

is satisfied, the only equilibria in the system are $(x_1, x_2)$ satisfying $\sin x_1 = 0$, $x_2 = 0$. One equilibrium corresponds to $x_1 = 0$, i.e., the pendulum is hanging vertically downward (stable), and the other equilibrium corresponds to $x_1 = \pi$, i.e., the vertically upward position (unstable). As $\dot{\theta}_a^2$ is increased beyond $g/l_p$, a supercritical pitchfork bifurcation of equilibria occurs [6]. The $(x_1, x_2) = (0, 0)$ equilibrium becomes unstable, and two other equilibria appear. These equilibria correspond to $\cos x_1 = \frac{g}{l_p \dot{\theta}_a^2}$, $x_2 = 0$.

We will now prove the stability of the equilibrium point at the origin for $\dot{\theta}_a$ satisfying (28), by constructing a Lyapunov function. Obviously the energy of this mechanical system can be used as a Lyapunov function, but since our purpose is to show that a Lyapunov function can be found using the SOS decomposition, we will assume that our knowledge is limited to the state equations describing the system and that we know nothing about the underlying energy.

Since the vector field (26)–(27) is not polynomial, a transformation to a polynomial vector field must be performed before we are able to construct a Lyapunov function using the SOS decomposition. For this purpose, introduce $u_1 = \sin x_1$ and $u_2 = \cos x_1$ to get:

$$\dot{x}_1 = x_2, \tag{29}$$

$$\dot{x}_2 = \dot{\theta}_a^2 u_1 u_2 - \frac{g}{l_p} u_1, \tag{30}$$

$$\dot{u}_1 = x_2 u_2, \tag{31}$$

$$\dot{u}_2 = -x_2 u_1. \tag{32}$$

In addition, we have the algebraic constraint

$$u_1^2 + u_2^2 - 1 = 0. \tag{33}$$

The whirling pendulum system will now be described by Equations (29)–(33). Notice that all the functions here are polynomial, so that Proposition 4 can be used to prove stability.

We will perform the analysis with the parameters of the system set at some fixed values. Assume that all the parameters except $g$ are equal to 1, and $g$ itself is equal to 10, for which condition (28) is satisfied. For a mechanical system like this, we expect that some trigonometric terms will be needed in the Lyapunov function. Thus we will try to find a Lyapunov function of the following form:

$$\begin{aligned} V &= a_1 x_2^2 + a_2 u_1^2 + a_3 u_2^2 + a_4 u_2 + a_5, \\ &= a_1 x_2^2 + a_2 \sin^2 x_1 + a_3 \cos^2 x_1 + a_4 \cos x_1 + a_5 \end{aligned} \tag{34}$$

where the $a_i$'s are the unknown coefficients. These coefficients must satisfy

$$a_3 + a_4 + a_5 = 0, \tag{35}$$

for $V$ to be equal to zero at $(x_1, x_2) = (0, 0)$. To guarantee that $V$ is positive definite, we search for $V$s that satisfy

$$V - \epsilon_1(1 - u_2) - \epsilon_2 x_2^2 \geq 0, \tag{36}$$

where $\epsilon_1$ and $\epsilon_2$ are positive constants (we set $\epsilon_1 \geq 0.1$, $\epsilon_2 \geq 0.1$). Positive definiteness holds as

$$\epsilon_1(1 - u_2) + \epsilon_2 x_2^2 = \epsilon_1(1 - \cos x_1) + \epsilon_2 x_2^2$$

is a positive definite function in the $(x_1, x_2)$–space (assuming all $x_1$ that differ by $2\pi$ are in the same equivalence class).

An example of Lyapunov function for this whirling pendulum system, found using the sum of squares procedure, is given by

$$V = 0.33445x_2^2 + 1.4615u_1^2 + 1.7959u_2^2 - 6.689u_2 + 4.8931.$$

### 4.3 Example 4.3: System with an Irrational Power Vector Field

Enzymatic reactions that are described by Michaelis-Menten type equations [7] usually contain terms with non-integer powers. Here we give an example of how such systems can be analyzed. Consider a simple one dimensional system:

$$\dot{x} = x^\alpha - 1, \quad x \in \mathbb{R}_+,$$

where $\alpha$ is a parameter. The linearisation of this system about the equilibrium $x = 1$ is $\dot{x} = \alpha x$ which implies that the system is locally stable for $\alpha < 0$. Let us make a transformation $y = x - 1$ to the above system to put its equilibrium at the origin:

$$\dot{y} = (y + 1)^\alpha - 1. \tag{37}$$

Further to this transformation, we introduce the transformation $z = (y + 1)^\alpha - 1$ and embed the system into a second order system with a polynomial vector field and an equality constraint that projects it back to 1-D:

$$\begin{aligned} \dot{y} &= z \\ \dot{z} &= \alpha \frac{(z+1)z}{y+1} \\ z &= (y+1)^\alpha - 1. \end{aligned} \tag{38}$$

The non-polynomial equality constraint (38) cannot be imposed in the sum of squares program in a similar manner as before. To proceed with the analysis, we will try to prove stability of the two dimensional system without the

equality constraint, but keeping in mind that the system is, at the end of the day, one-dimensional. We will attempt to prove stability for

$$\alpha - \alpha_h \leq 0 \tag{39}$$
$$-y + y_l \leq 0 \tag{40}$$
$$-z + z_l \leq 0. \tag{41}$$

We set $\alpha_h = -0.1$ and $y_l = -0.9$. This dictates that $z \geq (y_l + 1)^{\alpha_h} - 1 \triangleq z_l$. We search for a 4th order Lyapunov function in $y, z$ but we do not require $V$ to be positive definite in both $y$ and $z$, by constructing $\phi(y, z)$ in (21) appropriately. In particular the two Lyapunov conditions become:

$$V(y, z; \alpha) - \phi(y, z) \geq 0,$$
$$-\frac{\partial V}{\partial y}\dot{y} - \frac{\partial V}{\partial z}\dot{z} \leq 0,$$

for $\alpha, y, z$ satisfying (39)–(41), and additionally where

$$\phi(y, z) = \epsilon_1 y^2 + \epsilon_2 y^4 + \epsilon_3 z^2 + \epsilon_4 z^4,$$
$$\sum_{i=1}^{4} \epsilon_i \geq 0.01, \quad \epsilon_i \geq 0, \ \forall i = 1, \ldots, 4.$$

The inequality constraints (39)–(41) can be adjoined to the two conditions and a sum of squares program can be written using SOSTOOLS as in the previous examples. Indeed, such a Lyapunov function was constructed which allows for stability to be concluded.

### 4.4 Example 4.4: Diabatic Continuous Stirred Tank Reactor

Chemical reactors are the most important unit operation in a chemical process. In this section we consider the analysis of the dynamics of a perfectly mixed, diabatic, continuously stirred tank reactor (CSTR) [1]. We also assume a constant volume - constant parameter system for simplicity.

The reaction taking place in the CSTR is a first-order exothermic irreversible reaction $A \rightarrow B$. After balancing mass and energy, the reactor temperature $T$ and the concentration of species $A$ in the reactor $C_A$ evolve as follows:

$$\dot{C}_A = \frac{F}{V}(C_{A_f} - C_A) - k_0 e^{-\frac{\Delta E}{RT}} C_A \tag{42}$$
$$\dot{T} = \frac{F}{V}(T_f - T) - \frac{\Delta H}{\rho c_p} k_0 e^{-\frac{\Delta E}{RT}} C_A - \frac{UA}{V \rho c_p}(T - T_j) \tag{43}$$

where $F$ is the volumetric flow rate, $V$ is the reactor volume, $C_{A_f}$ is the concentration of $A$ in the freestream, $k_0$ is the pre-exponential factor of Arrhenius

**Table 1.** Parameter values for the CSTR.

| Parameter | Units | Nominal Value |
|---|---|---|
| $F/V$ | $\text{hr}^{-1}$ | 1 |
| $k_0$ | $\text{hr}^{-1}$ | 9703×3600 |
| $-\Delta H$ | kcal/kgmol | 5960 |
| $\Delta E$ | kcal/kgmol | 11843 |
| $\rho c_p$ | kcal/($\text{m}^3$ °C) | 500 |
| $T_f$ | °C | 25 |
| $C_{A_f}$ | kgmol/$\text{m}^3$ | 10 |
| $UA/V$ | kcal/($\text{m}^3$ °C hr) | 150 |
| $T_j$ | °C | 25 |

law, $\Delta E$ is the reaction activation energy, $R$ is the ideal gas constant, $T_f$ is the feed temperature, $-\Delta H$ is the heat of reaction (exothermic), $\rho$ is the density, $c_p$ is the heat capacity, $U$ is the overall heat transfer coefficient, $A$ is the area for heat exchange, and $T_j$ is the jacket temperature. For the analysis, we use the values shown in Table 1.

The equilibrium of the above system is given by $(C_{A_0}, T_0) = (8.5636, 311.171)$. We employ the following transformation: $x_1 = C_A/C_{A_0} - 1$, $x_2 = T/T_0 - 1$; this serves two purposes: firstly it moves the equilibrium to the origin, and secondly it rescales the state to avoid numerical ill-conditioning. The transformed system then becomes:

$$\dot{x}_1 = \frac{F}{V}\left(\frac{C_{A_f}}{C_{A_0}} - (x_1+1)\right) - k_0 e^{-\frac{\Delta E}{RT_0(x_2+1)}}(x_1+1)$$

$$\dot{x}_2 = \frac{F}{V}\left(\frac{T_f}{T_0} - (x_2+1)\right) - \frac{\Delta H C_{A_0}}{\rho c_p T_0} k_0 e^{-\frac{\Delta E}{RT_0(x_2+1)}}(x_1+1) - \frac{UA}{V\rho c_p}\left((x_2+1) - \frac{T_j}{T_0}\right)$$

Note that the system has an exponential term; the recasting will yield an indirect constraint, as discussed in Section 3. Define the state $x_3 = e^{\frac{\Delta E x_2}{RT_0(x_2+1)}} - 1$. Then an extra equation in the analysis would be

$$\dot{x}_3 = \frac{\Delta E}{RT_0(x_2+1)^2}(x_3+1)\dot{x}_2$$

under the constraint that $x_3 > -1$.

Then the full system, after we use the equilibrium relationship simplifies to:

$$\dot{x}_1 = -\frac{F}{V}x_1 - k_0 e^{-\frac{\Delta E}{RT_0}}(x_1x_3 + x_1 + x_3) \tag{44}$$

$$\dot{x}_2 = -\frac{F}{V}x_2 - \frac{\Delta H C_{A_0}}{\rho c_p T_0} k_0 e^{-\frac{\Delta E}{RT_0}}(x_1x_3 + x_1 + x_3) - \frac{UA}{V\rho c_p}x_2 \tag{45}$$

$$\dot{x}_3 = \frac{\Delta E}{RT_0(x_2+1)^2}(x_3+1)\dot{x}_2 \tag{46}$$

This system is now of the form (10)–(11), with $\tilde{x}_1 = (x_1, x_2)$ and $\tilde{x}_2 = x_3$. To proceed, we define the set $\mathcal{D}_1$ as:

$$\mathcal{D}_1 = \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| \leq \gamma_1, |x_2| \leq \gamma_2\}$$

and then define the set $\mathcal{D}_2$ as

$$\mathcal{D}_2 = \{x_3 \in \mathbb{R} : (x_3 - e^{\frac{-\Delta E\gamma_2}{RT_0(-\gamma_2+1)}} - 1)(x_3 - e^{\frac{\Delta E\gamma_2}{RT_0(\gamma_2+1)}} - 1) \leq 0\}$$

Then the system is ready for analysis as per Proposition 4. For $\gamma_1 = 0.12$ and $\gamma_2 = 0.05$ a quartic Lyapunov function can be constructed for the system described by Equations (44)–(46) using Proposition 4. Here the following $\phi(x)$ is used:

$$\phi(x) = \sum_{i=1}^{2}\sum_{j=2,4}\epsilon_{i,j}x_i^j + \sum_{j=1}^{4}\epsilon_{3,j}x_3^j,$$

with

$$\epsilon_{1,2} + \epsilon_{1,4} - 0.1 \geq 0$$
$$\epsilon_{2,2} + \epsilon_{2,4} + \epsilon_{3,1} + \epsilon_{3,2} + \epsilon_{3,3} + \epsilon_{3,4} - 0.1 \geq 0.$$

The level curves of the constructed Lyapunov function are shown in Figure 3.

## 5 Conclusions

In this chapter we have presented a methodology to analyze systems described by non-polynomial vector fields using the sum of squares decomposition and a recasting procedure. Using this recasting procedure, a non-polynomial system can be converted into a rational form. An extension of the Lyapunov theorem in conjunction with the sum of squares decomposition and semidefinite programming can then be used to investigate the stability of the recasted system, the result of which can be used to infer the stability of the original system. Some examples of systems whose vector fields contain radical, trigonometric, irrational power, and exponential terms have been presented to illustrate the use of the proposed approach.

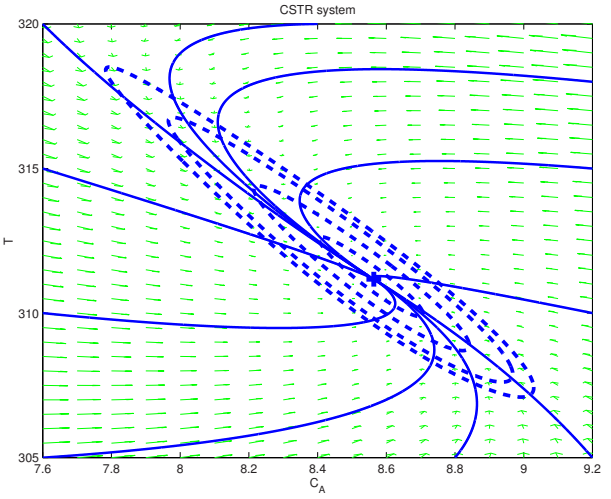



**Fig. 3.** Lyapunov function level curves for the system (42)–(43)

**Acknowledgements.**
The authors would like to acknowledge the input of Professors Michael Savageau and John C. Doyle.

## References

1. B. W. Bequette (1998). Process Dynamics. Modeling, Analysis and Simulation. Prentice Hall, NJ.
2. J. Bochnak, M. Coste, and M.-F. Roy (1998). Real Algebraic Geometry. Springer-Verlag, Berlin.
3. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia.
4. K. Furuta, M. Yamakita, and S. Kobayashi (1992). Swing-up control of inverted pendulum using pseudo-state feedback. Journal of Systems and Control Engineering, 206:263–269.
5. W. Hahn (1967). Stability of Motion. Springer-Verlag, NY.
6. J. Marsden and T. Ratiu (1999). Introduction to Mechanics and Symmetry. Springer-Verlag, NY, second edition.
7. J. D. Murray (1993). Mathematical Biology. Springer-Verlag, NY, second edition.
8. K. G. Murty and S. N. Kabadi (1987). Some NP-complete problems in quadratic and nonlinear programming. Mathematical Programming, 39:117–129.
9. A. Papachristodoulou and S. Prajna (2002). On the construction of Lyapunov functions using the sum of squares decomposition. Proc. IEEE Conf. on Decision and Control.

10. P. A. Parrilo (2000). Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. PhD thesis, Caltech, Pasadena, CA. Available at `www.control.ethz.ch/~parrilo/pubs/index.html`.
11. P. A. Parrilo and B. Sturmfels (1998). Minimizing polynomial functions. Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science.
12. S. Prajna, A. Papachristodoulou, and P. A. Parrilo (2002). Introducing SOSTOOLS: A general purpose sum of squares programming solver. Proc. IEEE Conf. on Decision and Control. Available at `www.cds.caltech.edu/sostools` and `www.aut.ee.ethz.ch/~parrilo/sostools`.
13. B. Reznick (2000). Some concrete aspects of Hilbert's 17th problem. Contemporary Mathematics, 253:251–272, AMS.
14. M. A. Savageau and E. O. Voit (1987). Recasting nonlinear differential equations as S-systems: a canonical nonlinear form. Mathematical Biosciences, 87(1):83–115.
15. J. F. Sturm (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 11–12:625–653. Available at `fewcal.kub.nl/sturm/software/sedumi.html`.
16. L. Vandenberghe and S. Boyd (1996). Semidefinite programming. SIAM Review, 38(1):49–95.
17. V. A. Yakubovich (1977). S-procedure in nonlinear control theory. Vestnik Leningrad University, 4(1):73–93. English translation.
18. V. Zubov (1964). Methods of A.M. Lyapunov and Their Application. P. Noordhoff Ltd, Groningen, The Netherlands.

# A Sum-of-Squares Approach to Fixed-Order $H_\infty$-Synthesis

C.W.J. Hol[1⋆] and C.W. Scherer[2]

[1] Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands. c.w.j.hol@dcsc.tudelft.nl
[2] Delft University of Technology, Delft Center for Systems and Control, Mekelweg 2, 2628 CD Delft, The Netherlands. c.w.scherer@dcsc.tudelft.nl

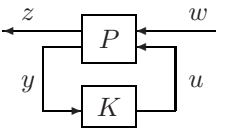
## 1 Introduction

Recent improvements of semi-definite programming solvers and developments on polynomial optimization have resulted in a large increase of the research activity on the application of the so-called sum-of-squares (SOS) technique in control. In this approach non-convex polynomial optimization programs are approximated by a family of convex problems that are relaxations of the original program [4, 22]. These relaxations are based on decompositions of certain polynomials into a sum of squares. Using a theorem of Putinar [28] it can be shown (under suitable constraint qualifications) that the optimal values of these relaxed problems converge to the optimal value of the original problem. These relaxation schemes have recently been applied to various non-convex problems in control such as Lyapunov stability of nonlinear dynamic systems [25, 5] and robust stability analysis [15].

In this work we apply these techniques to the fixed order or structured $\mathcal{H}_\infty$-synthesis problem. $\mathcal{H}_\infty$-controller synthesis is an attractive model-based control design tool which allows incorporation of modeling uncertainties in control design. We concentrate on $\mathcal{H}_\infty$-synthesis although the method can be applied to other performance specifications that admit a representation in terms of Linear Matrix Inequalities (LMI's). It is well-known that an $\mathcal{H}_\infty$-optimal full order controller can be computed by solving two algebraic Riccati equations [7]. However, the fixed order $\mathcal{H}_\infty$-synthesis problem is much more difficult. In fact it is one of the most important open problems in control engineering, in the sense that until now there do not yet exist fast and reliable methods to compute optimal fixed order controllers. As the basic setup we consider the closed-loop interconnection as shown below, where the linear system $P$ is the generalized plant and $K$ is a linear controller.

⋆This researcher is sponsored by Philips CFT

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 45–71, 2005.




Given $P$, we want to find a controller $K$ of a given order $n_c$ (independent of that of $P$) such that the closed-loop interconnection is internally (asymptotically) stable and such that the $\mathcal{H}_\infty$-norm of the closed-loop transfer function from $w$ to $z$ is minimized.

The resulting optimization problem is non-convex and difficult to solve. Various approaches have been presented in the literature based on sequential solution of LMI's [9, 2, 8, 20], nonlinear Semi-Definite Programs (SDP's) [23, 1, 18], Branch and Bound methods [35] and, more recently, polynomial optimization using SOS [14]. In this latter method a so-called central polynomial is a priori chosen such that a sufficient condition for the $\mathcal{H}_\infty$-performance can be formulated in terms of a positivity test of two matrix polynomials with coefficients that depend linearly on the controller variables. This linear dependence allows to find the variables satisfying these positivity constraints by LMI optimization. The method however depends crucially on the choice of this central polynomial which is in general difficult. Furthermore this technique cannot be straightforwardly extended to MIMO (Multiple Input Multiple Output) $\mathcal{H}_\infty$-optimal controller synthesis. In contrast, the method to be discussed below can directly be applied to MIMO synthesis.

The main result of this paper is the construction of a sequence of SOS polynomial relaxations

- that require the solution of LMI problems whose size grows only quadratically in the number of states and
- whose optimal value converge from below to the fixed-order $\mathcal{H}_\infty$ performance.

The computation of *lower bounds* allows to add a stopping criterion to the algorithms mentioned above with a guaranteed bound on the difference of the performance of the computed controller and the optimal fixed order $\mathcal{H}_\infty$ performance. This is important since, except for the branch and bound method, these algorithms can in general not guarantee convergence to the globally optimal solution.

A trivial lower bound on the fixed order performance is, of course, the optimal performance achievable with a controller of the same order as the plant. Boyd and Vandenberghe [3] proposed lower bounds based on convex relaxations of the fixed order synthesis problem. These lower bounds cannot be straightforwardly improved to reduce the gap to the optimal fixed order performance. For our sequence of relaxations this gap is systematically reduced to zero.

After the problem formulation in Section 2, we show in Section 3 how a suitable matrix sum-of-squares relaxation technique can be directly applied to the non-convex semi-definite optimization problem resulting from the bounded real lemma. We will prove that the values of these relaxations converge to the optimal value. Although this convergence property is well-known for polynomial problems with scalar constraints [22], to the best of our knowledge this result is new for matrix-valued inequalities. (During the writing of this paper we became aware of the independent recent work of Kojima [24], that presents the same result with a different proof). This convergence result is of value for a variety of matrix-valued optimization problems. Examples in control are input-output selection, where the integer constraints of type $p \in \{0, 1\}$ are replaced by a quadratic constraint $p(p-1) = 0$, and spectral factorization of multidimensional transfer functions to asses dissipativity of linear shift-invariant distributed systems [26]. Here, our goal is to apply it to the fixed order $\mathcal{H}_\infty$ synthesis problem. Unfortunately, for plants with high state-dimension this direct technique leads to an unacceptable complexity. As the main reason, the resulting relaxations involve the search for an SOS polynomial in *all* variables, including the Lyapunov matrix in the bounded real lemma inequality constraint. Therefore, the size of the LMI relaxations grows exponentially in the number of state-variables.

In Section 4 we describe how to overcome this deficiency by constructing a relaxation scheme without the exponential growth in the state dimension through two-fold sequential dualization. First we dualize in the variables that grow with the state dimension, which leads to the re-formulation of the fixed order synthesis problem as a robust analysis problem with the controller variables as parametric uncertainty. On the one hand, this allows to apply the wide spectrum of robust analysis techniques to the fixed order controller design problem. On the other hand, robustness has to be verified only with respect to the small number of controller parameters which is the essence of keeping the growth of the relaxation size in the state-dimension polynomial.

The purpose of Section 5 is to discuss a novel approach to solve robust LMI problems based on SOS matrices, including a direct and compact description of the resulting linear SDP's with full flexibility in the choice of the underlying monomial basis. This leads to an asymptotically exact family of LMI relaxations for computing lower bounds on the optimal fixed-order $\mathcal{H}_\infty$-norm whose size only grows quadratically in the dimension of the system state. We will reveal as well that existing results based on straightforward scalarization techniques fail to guarantee these growth properties.

Our technique is appropriate for the design of controllers with a few decision variables (such as Proportional Integral Derivative (PID) controllers) for plants with moderate Mc-Millan degree. In Section 6 we apply the method to the fixed order $\mathcal{H}_\infty$-synthesis problem on two systems: an academic model with Mc-Millan degree 4 and a model of an active suspension system which has a generalized plant of a Mc-Millan degree 27.

## 2 Fixed-Order $H_\infty$ Controller Synthesis

Consider the $\mathcal{H}_\infty$-reduced order synthesis problem with a closed-loop system described by $A(p)$, $B(p)$, $C(p)$ and $D(p)$, where $p$ parameterizes the to-be-constructed controller and varies in the compact set $\mathcal{P}$. Compactness can, for instance, be realized by restricting the controller variables to a Euclidean ball

$$\mathcal{P} := \{p \in \mathbb{R}^{n_p} \mid \|p\| \leq M\}. \tag{1}$$

In practice, most structured control problems have large state dimension and few controller variables that enter affinely in the closed-loop state space description. Suppose the generalized plant of order $n$ admits the state space description

$$\left(\begin{array}{c} \dot{x} \\ \hline z \\ y \end{array}\right) = \left(\begin{array}{c|cc} A^{\rm ol} & B_1^{\rm ol} & B_2^{\rm ol} \\ \hline C_1^{\rm ol} & D_{11}^{\rm ol} & D_{12}^{\rm ol} \\ C_2^{\rm ol} & D_{21}^{\rm ol} & 0 \end{array}\right) \left(\begin{array}{c} x \\ \hline w \\ u \end{array}\right),$$

where $(\cdot)^{\rm ol}$ stands for 'open loop' and $A^{\rm ol} \in \mathbb{R}^{n\times n}$, $B_1^{\rm ol} \in \mathbb{R}^{n\times m_1}$, $B_2^{\rm ol} \in \mathbb{R}^{n\times m_2}$, $C_1^{\rm ol} \in \mathbb{R}^{p_1\times n}$, $C_2^{\rm ol} \in \mathbb{R}^{p_2\times n}$, $D_{11}^{\rm ol} \in \mathbb{R}^{p_1\times m_1}$, $D_{12}^{\rm ol} \in \mathbb{R}^{p_1\times m_2}$ and $D_{21}^{\rm ol} \in \mathbb{R}^{p_2\times m_1}$. For simplicity we assume the direct-feedthrough term $D_{22}^{\rm ol}$ to be zero. In Remark 7 we will discuss how our method can be applied to control problems with nonzero $D_{22}^{\rm ol}$. Now consider, for instance, a PID-controller described by

$$k(p) = p_1 + p_2\frac{1}{s} + p_3\frac{s}{\tau s+1},$$

which admits the state space realization

$$\left(\begin{array}{c|c} A_K(p) & B_K(p) \\ \hline C_K(p) & D_K(p) \end{array}\right) := \left(\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & -\frac{1}{\tau} & \frac{1}{\tau} \\ \hline p_2 & -\frac{p_3}{\tau} & p_1 + \frac{p_3}{\tau} \end{array}\right)$$

and suppose that we want to find the optimal proportional, integral and derivative gains $p_1$, $p_2$ and $p_3$ respectively. This structure has been used by Ibaraki and Tomizuka [19] for $\mathcal{H}_\infty$-optimal PID tuning of a hard-disk drive using the cone complementarity method [9]. See also Grassi and Tsakalis [11, 12] and Grimble and Johnson [13] for $\mathcal{H}_\infty$ and LQG-optimal PID-tuning respectively. Interconnecting the plant with the PID-controller yields a closed-loop state-space representation with matrices

$$\left(\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right) = \left(\begin{array}{cc|c} A^{\rm ol} + B_2^{\rm ol}D_K(p)C_2^{\rm ol} & B_2^{\rm ol}C_K(p) & B_1^{\rm ol} + B_2^{\rm ol}D_K(p)D_{21}^{\rm ol} \\ B_K(p)C_2^{\rm ol} & A_K(p) & B_K(p)D_{21}^{\rm ol} \\ \hline C_1^{\rm ol} + D_{12}^{\rm ol}D_K(p)C_2^{\rm ol} & D_{12}^{\rm ol}C_K(p) & D_{11}^{\rm ol} + D_{12}^{\rm ol}D_K(p)D_{21}^{\rm ol} \end{array}\right)$$

which depend affinely on $p$. We intend to solve the fixed order $\mathcal{H}_\infty$-synthesis problem

$$\inf_{p\in\mathcal{P},\, A(p)\,\text{stable}} \|D(p) + C(p)(sI - A(p))^{-1}B(p)\|_\infty^2.$$

Due to the bounded real lemma we can solve instead the equivalent problem

$$\begin{aligned} &\text{infimize} && \gamma \\ &\text{subject to} && p \in \mathcal{P}, \quad X \in \mathcal{S}^n, \quad X \succ 0 \\ & && B_\infty(X, p, \gamma) \prec 0 \end{aligned} \tag{2}$$

where $\mathcal{S}^n$ denotes the set of real symmetric $n \times n$ matrices and

$$B_\infty(X,p,\gamma) := \begin{pmatrix} A(p)^T X + XA(p) & XB(p) \\ B(p)^T X & -\gamma I \end{pmatrix} + \begin{pmatrix} C(p)^T \\ D(p)^T \end{pmatrix} \begin{pmatrix} C(p) & D(p) \end{pmatrix}.$$

Let $t^p_{\text{opt}}$ denote the optimal value of (primal) problem (2). Due to the bilinear coupling of $X$ and $p$, this problem is a non-convex Bilinear Matrix Inequality problem (BMI). The number of bilinearly coupled variables is $\frac{1}{2}(n + n_c)(n + n_c + 1) + n_p$ (with $n_p$ denoting the number of free controller variables) which grows quadratically with $n$. However it has a linear objective and matrix valued polynomial constraints. It will be shown in the next section that we can compute a sequence of SOS relaxations of this problem that converges from below to the optimal value.

## 3 A Direct Polynomial SDP-Approach

In this section we present an extension of the scalar polynomial optimization by SOS decompositions [22] to optimization problems with scalar polynomial objective and nonlinear polynomial semi-definite constraints. We formulate the relaxations in terms of Lagrange duality with SOS polynomials as multipliers which seems a bit more straightforward than the corresponding dual formulation based on the problem of moments [22].

### 3.1 Polynomial Semi-definite Programming

For $x \in \mathbb{R}^n$ let $f(x)$ and $G(x)$ denote scalar and symmetric-matrix-valued polynomials in $x$, and consider the following polynomial semi-definite optimization problem with optimal value $d_{\text{opt}}$:

$$\begin{aligned} &\text{infimize} && f(x) \\ &\text{subject to} && G(x) \preceq 0 \end{aligned} \tag{3}$$

Since multiple SDP-constraints can be easily collected into one single SDP-constraint by diagonal augmentation, it is clear that (2) is captured by this general formulation.

With any matrix $S \succeq 0$, the value $\inf_{x\in\mathbb{R}^{n_x}} f(x) + \langle S, G(x)\rangle$ is a lower bound for $d_{\rm opt}$ by standard weak duality. However, not even the maximization of this lower bound over $S \succeq 0$ allows to close the duality gap due to non-convexity of the problem. This is the reason for considering, instead, Lagrange multiplier matrices $S(x) \succeq 0$ which are polynomial functions of $x$. Still $\inf_{x\in\mathbb{R}^{n_x}} f(x)+\langle S(x), G(x)\rangle$ defines a lower bound of $d_{\rm opt}$, and the best lower bound that is achievable in this fashion is given by the supremal $t$ for which there exists a polynomial matrix $S(x) \succeq 0$ such that

$$f(x) + \langle S(x), G(x)\rangle - t > 0 \quad \text{for all} \quad x \in \mathbb{R}^{n_x}.$$

In order to render the determination of this lower bound computational we introduce the following concept. A symmetric matrix-valued $n_G \times n_G$-polynomial matrix $S(x)$ is said to be a (matrix) sum-of-squares (SOS) if there exists a (not necessarily square and typically tall) polynomial matrix $T(x)$ such that

$$S(x) = T(x)^T T(x).$$

If $T_j(x)$, $j = 1, \ldots, q$ denote the rows of $T(x)$, we infer

$$S(x) = \sum_{j=1}^{q} T_j(x)^T T_j(x).$$

If $S(x)$ is a scalar then $T_j(x)$ are scalars which implies $S(x) = \sum_{j=1}^{q} T_j(x)^2$. This motivates our terminology since we are dealing with a generalization of classical scalar SOS representations. Very similar to the scalar case, every SOS matrix is globally positive semi-definite, but the converse is not necessarily true.

Let us now just replace all inequalities in the above derived program for the lower bound computations by the requirement that the corresponding polynomials or polynomial matrices are SOS. This leads to the following optimization problem:

$$\begin{aligned} &\text{supremize} \;\; t \\ &\text{subject to} \;\; S(x) \text{ and } f(x) + \langle S(x), G(x)\rangle - t \text{ are SOS} \end{aligned} \tag{4}$$

If fixing upper bounds on the degree of the SOS matrix $S(x)$, the value of this problem can be computed by solving a standard linear SDP as will be seen in Section 5. In this fashion one can construct a family of LMI relaxations for computing increasingly improving lower bounds. Under a suitable constraint qualification, due to Putinar for scalar problems, it is possible to prove that the value of (4) actually equals $d_{\rm opt}$. To the best of our knowledge, the generalization to matrix valued problems as formulated in the following result has, except for the recent independent work of Kojima [24], not been presented anywhere else in the literature.

**Theorem 1.** *Let $d_{opt}$ be the optimal solution of (3) and suppose the following constraint qualification holds true: There exists some $r > 0$ and some SOS matrix $R(x)$ such that*

$$r - \|x\|^2 + \langle R(x), G(x)\rangle \quad \textit{is SOS.} \tag{5}$$

*Then the optimal value of (4) equals $d_{opt}$.*

*Proof.* The value of (4) is not larger than $d_{\text{opt}}$. Since trivial for $d_{\text{opt}} = \infty$, we assume that $G(x) \preceq 0$ is feasible. Choose any $\epsilon > 0$ and some $\hat{x}$ with $G(\hat{x}) \preceq 0$ and $f(\hat{x}) \leq d_{\text{opt}} + \epsilon$. Let us now suppose that $S(x)$ and $f(x) + \langle S(x), G(x)\rangle - t$ are SOS. Then

$$d_{\text{opt}} + \epsilon - t \geq f(\hat{x}) - t \geq f(\hat{x}) + \langle S(\hat{x}), G(\hat{x})\rangle - t \geq 0$$

and thus $d_{\text{opt}} + \epsilon \geq t$. Since $\epsilon$ was arbitrary we infer $d_{\text{opt}} \geq t$.

To prove the converse we first reveal that, due to the constraint qualification, we can replace $G(x)$ by $\hat{G}(x) = \text{diag}(G(x), \|x\|^2 - r)$ in both (3) and (4) without changing their values. Indeed if $G(x) \preceq 0$ we infer from (5) that $r - \|x\|^2 \geq r - \|x\|^2 + \langle R(x), G(x)\rangle \geq 0$. Therefore the extra constraint $\|x\|^2 - r \leq 0$ is redundant for (3). We show redundancy for (4) in two steps. If $S(x)$ and $f(x) - t + \langle S(x), G(x)\rangle$ are SOS we can define the SOS matrix $\hat{S}(x) = \text{diag}(S(x), 0)$ to conclude that $f(x) - t + \left\langle \hat{S}(x), \hat{G}(x) \right\rangle$ is SOS (since it just equals $f(x) - t + \langle S(x), G(x)\rangle$). Conversely suppose that $\hat{S}(x) = \hat{T}(x)^T \hat{T}(x)$ and $\hat{t}(x)^T \hat{t}(x) = f(x) - t + \left\langle \hat{S}(x), \hat{G}(x) \right\rangle$ are SOS. Partition $\hat{T}(x) = (T(x)\ u(x))$ according to the columns of $\hat{G}(x)$. With the SOS polynomial $v(x)^T v(x) = r - \|x\|^2 + \langle R(x), G(x)\rangle$ we infer

$$\begin{aligned}
&\hat{t}(x)^T \hat{t}(x) = f(x) - t + \left\langle T(x)^T T(x), G(x) \right\rangle + u(x)^T u(x)(\|x\|^2 - r) = \\
&= f(x) - t + \left\langle T(x)^T T(x), G(x) \right\rangle + u(x)^T u(x)(\langle R(x), G(x)\rangle - v(x)^T v(x)) = \\
&= f(x) - t + \left\langle T(x)^T T(x) + u(x)^T u(x) R(x), G(x) \right\rangle - u(x)^T u(x) v(x)^T v(x).
\end{aligned}$$

With $R(x) = R_f(x)^T R_f(x)$ we now observe that

$$S(x) := T(x)^T T(x) + u(x)^T u(x) R(x) = \begin{pmatrix} T(x) \\ u(x) \otimes R_f(x) \end{pmatrix}^T \begin{pmatrix} T(x) \\ u(x) \otimes R_f(x) \end{pmatrix}$$

and

$$s(x) := \hat{t}(x)^T \hat{t}(x) + u(x)^T u(x) v(x)^T v(x) = \begin{pmatrix} \hat{t}(x) \\ u(x) \otimes v(x) \end{pmatrix}^T \begin{pmatrix} \hat{t}(x) \\ u(x) \otimes v(x) \end{pmatrix}$$

are SOS. Due to $f(x) - t + \langle S(x), G(x)\rangle = s(x)$ the claim is proved.

Hence from now one we can assume without loss of generality that there exists a standard unit vector $v_1$ with

$$v_1^T G(x) v_1 = \|x\|^2 - r. \tag{6}$$

Let us now choose a sequence of unit vectors $v_2, v_3, \ldots$ such that $v_i$, $i = 1, 2, \ldots$ is dense in the Euclidean unit sphere, and consider the family of scalar polynomial optimization problems

$$\begin{aligned} &\text{infimize} \;\; f(x) \\ &\text{subject to} \;\; v_i^T G(x) v_i \le 0, \quad i = 1, \ldots, N \end{aligned} \tag{7}$$

with optimal values $d_N$. Since any $x$ with $G(x) \preceq 0$ is feasible for (7), we infer $d_N \le d_{\rm opt}$. Moreover it is clear that $d_N \le d_{N+1}$ which implies $d_N \to d_0 \le d_{\rm opt}$ for $N \to \infty$. Let us prove that $d_0 = d_{\rm opt}$. Due to (6) the feasible set of (7) is contained in $\{x \in \mathbb{R}^{n_x} \mid \|x\|^2 \le r\}$ and hence compact. Therefore there exists an optimal solution $x_N$ of (7), and we can choose a subsequence $N_\nu$ with $x_{N_\nu} \to x_0$. Hence $d_0 = \lim_{\nu\to\infty} d_{N_\nu} = \lim_{\nu\to\infty} f(x_{N_\nu}) = f(x_0)$. Then $d_0 = d_{\rm opt}$ follows if we can show that $G(x_0) \preceq 0$. Otherwise there exists a unit vector $v$ with $\epsilon := v^T G(x_0) v > 0$. By convergence there exists some $K$ with $\|G(x_{N_\nu})\| \le K$ for all $\nu$. By density there exists a sufficiently large $\nu$ such that $K\|v_i - v\|^2 + 2K\|v_i - v\| < \epsilon/2$ for some $i \in \{1, \ldots, N_\nu\}$. We can increase $\nu$ to guarantee $v^T G(x_{N_\nu}) v \ge \epsilon/2$ and we arrive at

$$\begin{aligned} 0 \ge v_i^T G(x_{N_\nu}) v_i = \\ = (v_i - v)^T G(x_{N_\nu})(v_i - v) + 2v^T G(x_{N_\nu})(v_i - v) + v^T G(x_{N_\nu}) v \ge \\ \ge -K\|v_i - v\|^2 - 2K\|v_i - v\| + \epsilon/2 > 0, \end{aligned}$$

a contradiction.

Let us finally fix any $\epsilon > 0$ and choose $N$ with $d_N \ge d_{\rm opt} - \epsilon/2$. This implies $f(x) - d_{\rm opt} + \epsilon > 0$ for all $x$ with $v_i^T G(x) v_i \le 0$ for $i = 1, \ldots, N$. Due to (6) we can apply Putinar's scalar representation result [28] to infer that there exist polynomials $t_i(x)$ for which

$$f(x) - d_{\rm opt} + \epsilon + \sum_{i=1}^{N} t_i(x)^T t_i(x) v_i^T G(x) v_i \;\; \text{is SOS}. \tag{8}$$

With the SOS matrix

$$S_N(x) := \sum_{i=1}^{N} v_i t_i(x)^T t_i(x) v_i^T = \begin{pmatrix} t_1(x) v_1^T \\ \vdots \\ t_N(x) v_N^T \end{pmatrix}^T \begin{pmatrix} t_1(x) v_1^T \\ \vdots \\ t_N(x) v_N^T \end{pmatrix}$$

we conclude that $f(x) - d_{\rm opt} + \epsilon + \langle S_N(x), G(x) \rangle$ equals (8) and is thus SOS. This implies that the optimal value of (4) is at least $d_{\rm opt} - \epsilon$, and since $\epsilon > 0$ was arbitrary the proof is finished. ■

Theorem 1 is a natural extension of a theorem of Putinar [28] for scalar polynomial problems to polynomial SDP's. Indeed, Lasserre's approach [22] for minimizing $f(x)$ over scalar polynomial constraints $g_i(x) \leq 0$, $i = 1, \ldots, m$, is recovered with $G(x) = \mathrm{diag}(g_1(x), \ldots, g_m(x))$. Moreover the constraint qualification in Theorem 1 is a natural generalization of that used by Schweighofer [33].

*Remark 1.* It is a direct consequence of Theorem 1 that, as in the scalar case [32], the constraint qualification (5) can be equivalently formulated as follows: there exist an SOS matrix $R(x)$ and an SOS polynomial $s(x)$ such that

$$\{x \in \mathbb{R}^{n_x} \mid \langle R(x), G(x)\rangle - s(x) \leq 0\} \text{ is compact.}$$

### 3.2 Application to the $\mathcal{H}_\infty$ fixed order control problem

The technique described above can directly be applied to (2), except that the constraint qualification is not necessarily satisfied. This can be resolved by appending a bounding inequality $X \preceq M_X I$ for some large value $M_X > 0$. An SOS relaxation of the resulting BMI problem is formulated with

$$G_1(X,p,\gamma) := -X, \quad G_2(X,p,\gamma) := B_\infty(X,p,\gamma),$$
$$G_3(X,p,\gamma) := X - M_X I, \quad G_4(X,p,\gamma) := \|p\|^2 - M$$

as follows

$$\begin{aligned} &\text{supremize:} && \gamma \\ &\text{subject to:} && \gamma + \sum_{i=1}^{4} \langle S_i(X,p,\gamma), G_i(X,p,\gamma)\rangle \quad \text{is SOS} \\ & && S_i(X,p,\gamma) \ \text{ is SOS}, \quad i = 1, \ldots, 4. \end{aligned} \tag{9}$$

It has already been indicated that this problem is easily translated into a linear SDP if we impose a priori bounds on the degrees of all SOS matrix polynomials. However, as the main trouble, this technique suffers from large number of variables for higher order relaxations, especially when the order of the underlying system state is large. Since $S_1$, $S_2$ and $S_3$ are polynomials in $X$, the size of the relaxation grows *exponentially* with the order of the system. In our numerical experience the size of the LMI problems of the SOS relaxations grows so fast that good lower bounds can be only computed for systems with state dimension up to about 4. Therefore it is crucial to avoid the need for constructing SOS polynomials in the Lyapunov variable $X$. As the second main contribution of this paper, we reveal in the next section how to overcome this deficiency.

## 4 Conversion to Robustness Analysis

For translating the nonlinear synthesis problem in (2) to an equivalent robustness analysis problem, the key idea is to apply *partial* Lagrange dualization [17]: Fix the controller variables $p$ and dualize with respect to the Lyapunov variable $X$. We will show that one is required to determine parameter-dependent dual variables, in full analogy to computing parameter-dependent Lyapunov function for LPV systems. As the main advantage, this reformulation allows us to suggest novel SOS relaxations that grow exponentially only in the number of controller (or uncertain) parameters $p$ and that can be shown to grow only *quadratically* in the number of the system states, in stark contrast to the relaxations of Section 3.2.

### 4.1 Partial Dualization

In this section we need the additional assumption that $(A(p), B(p))$ is controllable for every $p \in \mathcal{P}$. For fixed $p = p_0 \in \mathcal{P}$, (2) is an LMI problem in $X$ and $\gamma$:

$$\begin{aligned} &\text{infimize} \;\; \gamma \\ &\text{subject to} \;\; X \in \mathcal{S}^n, \qquad X \succ 0, \qquad B_\infty(X, p_0, \gamma) \prec 0. \end{aligned} \tag{10}$$

Let us partition the dual variable $Z$ for the constraint $B_\infty(X, p, \gamma) \prec 0$ in (2) as

$$Z = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{pmatrix}. \tag{11}$$

Then the Langrange dual reads as follows:

$$\begin{aligned} \text{supremize} \;\; & \operatorname{Tr}\left( \begin{bmatrix} C(p_0) & D(p_0) \end{bmatrix} Z \begin{bmatrix} C(p_0) & D(p_0) \end{bmatrix}^T \right) \\ \text{subject to} \;\; & A(p_0)Z_{11} + Z_{11}A(p_0)^T + B(p_0)Z_{12}^T + Z_{12}B(p_0)^T \succeq 0 \\ & \operatorname{Tr}(Z_{22}) \leq 1, \qquad Z \succeq 0. \end{aligned} \tag{12}$$

Let $t^d_{\text{opt}}(p_0)$ denote the dual optimal value of (12). Note that (12) is strictly feasible for all $p_0 \in \mathcal{P}$ as is shown in Appendix A. This implies $t^d_{\text{opt}}(p) = t^p_{\text{opt}}(p)$ and, as a consequence, it allows to draw the following conclusion. Given any $t \in \mathbb{R}$ suppose that the function $Z(p)$ satisfies

$$\operatorname{Tr}\left( \begin{bmatrix} C(p) & D(p) \end{bmatrix} Z(p) \begin{bmatrix} C(p) & D(p) \end{bmatrix}^T \right) > t, \tag{13}$$

$$A(p)Z_{11}(p) + Z_{11}(p)A(p)^T + B(p)Z_{12}(p)^T + Z_{12}(p)B^T(p) \succ 0, \tag{14}$$

$$\operatorname{Tr}(Z_{22}(p)) < 1, \qquad Z(p) \succ 0, \tag{15}$$

for all $p \in \mathcal{P}$. Then it is clear that $t^d_{\text{opt}}(p) \geq t$ and hence $t^p_{\text{opt}}(p) \geq t$ hold for all $p \in \mathcal{P}$. Therefore $t$ is a lower bound on the best achievable controller performance. It is thus natural to maximize $t$ over some class of functions $Z(.)$ in order to determine tight lower bounds on the value of (2). Our construction allows to show that this lower bound is actually tight if optimizing over matrix polynomials $Z(.)$.

**Theorem 2.** *Let $\gamma_{opt}$ be the optimal solution to* (2)*. Let $t_{opt}$ be the supremal $t$ for which there exists a polynomial matrix $Z(p) \in \mathcal{S}^{n+m_1}$ satisfying* (13)-(15) *for all $p \in \mathcal{P}$. Then $\gamma_{opt} = t_{opt}$.*

*Proof.* We have already seen that $\gamma_{\text{opt}} \geq t_{\text{opt}}$. Now suppose $\gamma_{\text{opt}} \geq t_{\text{opt}} + \epsilon$ for some $\epsilon > 0$. For any fixed $p_0 \in P$, the optimal value of (10) and hence that of (12) are not smaller than $\gamma_{\text{opt}}$. Since (12) is strictly feasible there exists $Y^0$ (partitioned as (11)) with

$$\begin{aligned} \text{Tr}\left(\begin{bmatrix} C(p_0) & D(p_0) \end{bmatrix} Y^0 \begin{bmatrix} C(p_0) & D(p_0) \end{bmatrix}^T\right) - \epsilon/2 > t_{\text{opt}}, \\ A(p_0)Y^0_{11} + Y^0_{11}A(p_0)^T + B(p_0){Y^0_{12}}^T + Y^0_{12}B^T(p_0) \succ 0, \\ \text{Tr}(Y^0_{22}) < 1, \qquad Y^0 \succ 0. \end{aligned}$$

Since the inequalities are strict and $\mathcal{P}$ is compact, we can use a partition of unity argument [29] to show that there actually exists a *continuous* function $Y(p)$ such that

$$\text{Tr}\left(\begin{bmatrix} C(p) & D(p) \end{bmatrix} Y(p) \begin{bmatrix} C(p) & D(p) \end{bmatrix}^T\right) - \epsilon/4 > t_{\text{opt}}, \quad (16)$$

$$A(p)Y_{11}(p) + Y_{11}(p)A(p)^T + B(p)Y_{12}(p)^T + Y_{12}(p)B^T(p) \succ 0, \quad (17)$$

$$\text{Tr}(Y_{22}(p)) < 1, \qquad Y(p) \succ 0, \quad (18)$$

for all $p \in \mathcal{P}$. Due to the Stone-Weisterstrass theorem about the approximation of continuous functions by polynomials on compacta, we can even choose $Y(p)$ to be a matrix polynomial. This allows to conclude $t_{\text{opt}} \geq t_{\text{opt}} + \epsilon/4$, a contradiction which finishes the proof. ∎

In actual computations we optimize over functions $Z(\cdot)$ belonging to an increasing sequence of finite-dimensional subspaces of matrix-valued polynomials. Then the difference of the computed lower bound to the actual optimal $\mathcal{H}_\infty$ performance is non-decreasing. If we restrict the search to a subspace of degree bounded matrix polynomials, and if we let the bound on the degree grow to infinity, Theorem 2 guarantees that the corresponding lower bounds converge from below to the globally optimal $\mathcal{H}_\infty$ performance.

We have thus reduced the $\mathcal{H}_\infty$ synthesis problem to a robust analysis problem with complicating variables $p$ and polynomial robustness certificates

$Z(p)$. In Section 5 we will discuss how (13)-(15) can be relaxed to standard LMI constraints via suitable SOS tests.

*Remark 2.* The proposed partial dualization technique is not at all restricted to fixed-order $\mathcal{H}_\infty$ optimal control. Straightforward variations do apply to a whole variety of other interesting problems, such as designing structured controllers for any performance criterion that admits an analysis LMI representation (as e.g. general quadratic performance, $H_2$-performance or placement of closed-loop poles in LMI regions [6]).

*Remark 3.* We require the controller parameters to lie in a compact set in order to be able to apply the theorem of Stone-Weierstrass. From a practical point of view this is actually not restrictive since the controller parameters have to be restricted in size for digital implementation. Moreover one can exploit the flexibility in choosing the set $\mathcal{P}$ in order to incorporate the suggested lower bound computations in branch-and-bound techniques.

*Remark 4.* The controllability assumption is needed to prove that the dual (12) is strictly feasible for all $p \in \mathcal{P}$. Controllability can be verified by a Hautus test: $(A(p), B(p))$ is controllable for all $p \in \mathcal{P}$ if and only if

$$P_{\mathrm{H}}(\lambda, p) := \Big( A(p) - \lambda I \;\; B(p) \Big) \text{ has full row rank for all } \lambda \in \mathbb{C},\ p \in \mathcal{P}. \quad (19)$$

This property can be verified by the method described in Section 3. Indeed suppose $K \in \mathbb{R}$ is chosen with $\|A(p)\| \leq K$ for all $p \in \mathcal{P}$ (with $\|\cdot\|$ denoting the spectral norm). Then (19) holds true if and only if the real-valued polynomial

$$\begin{aligned} F_{\mathrm{H}}(a, b, p) :&= \left|\det \left(P_{\mathrm{H}}(a + bi, p) P_{\mathrm{H}}(a + bi, p)^*\right)\right|^2 \\ &= \det \left(P_{\mathrm{H}}(a + bi, p) P_{\mathrm{H}}(a + bi, p)^*\right)^* \det \left(P_{\mathrm{H}}(a + bi, p) P_{\mathrm{H}}(a + bi, p)^*\right) \end{aligned}$$

is strictly positive on $[-K, K] \times [-K, K] \times \mathcal{P}$. This can be tested with SOS decompositions, provided that $\mathcal{P}$ has a representation that satisfies (5). The upper bound $K$ on the spectral norm of $A$ on $\mathcal{P}$ can also be computed with SOS techniques.

*Remark 5.* The utmost right constraint in (14) ($Z(p) \succ 0$ for all $p \in \mathcal{P}$) can be replaced by the (generally much) stronger condition

$$Z(p) \text{ is SOS in } p.$$

As we will see in Section 5.6 this may reduce the complexity of our relaxation problems. Theorem 2 is still true after the replacement, since for any matrix-valued polynomial $Y(p)$ that satisfies (13)-(15), we can find a unique matrix valued function $R(p)$ on $\mathcal{P}$ that is the Cholesky factor of $Z(p)$ for all $p \in \mathcal{P}$.

Furthermore $R(p)$ is continuous on $\mathcal{P}$ if $Z(p)$ is, because the Cholesky factor of a matrix can be computed by a sequence of continuity preserving operations on the coefficients of the matrix [10]. Again by Weierstrass' theorem there exists an approximation of the continuous $R(p)$ on $\mathcal{P}$ by a polynomial $\tilde{R}(p)$ such $Z(p) := \tilde{R}(p)^T\tilde{R}(p)$ satisfies (13)-(15). The constructed matrix-valued polynomial $Z(p)$ is indeed SOS.

### 4.2 Finite-Dimensional Approximation

Suppose that $Z_j : \mathbb{R}^{n_p} \mapsto \mathcal{S}^{n+m_1}$, $j = 1, 2, \ldots, N$, is a set of linearly independent symmetric-valued polynomial functions in $p$ (such as a basis for the real vector space of all symmetric matrix polynomials of degree at most $d$). Let us now restrict the search of $Z(.)$ in Theorem 2 to the subspace

$$\mathcal{Z}_N := \left\{ Z(.,z) \mid Z(.,z) := \sum_{j=1}^{N} z_j Z_j(.),\ z = (z_1, \ldots, z_N) \in \mathbb{R}^N \right\}. \tag{20}$$

Then (13)-(15) are polynomial inequalities in $p$ that are affine in the coefficients $z$ for the indicated parameterization of the elements $Z(.,z)$ in $\mathcal{Z}_N$. With $y := \mathrm{col}(t, z)$, $c := \mathrm{col}(1, 0_{n_z})$ and

$$F(p,y) := \mathrm{diag}(F_{11}(p,z) - t, F_{22}(p,z), Z(p,z), 1 - \mathrm{Tr}(Z_{22}(p,z))) \tag{21}$$

where

$$F_{11}(p,z) := \mathrm{Tr}\left( \begin{bmatrix} C(p) & D(p) \end{bmatrix} Z(p,z) \begin{bmatrix} C(p) & D(p) \end{bmatrix}^T \right),$$

$$F_{22}(p,z) := Z_{11}(p,z)A(p)^T + A(p)Z_{11}(p,z) + Z_{12}(p,z)B(p)^T + B(p)Z_{12}^T(p,z)^T$$

the problem to be solved can be compactly written as follows:

$$\begin{aligned} &\text{supremize} \ \ c^T y \\ &\text{subject to} \ \ F(p,y) \succ 0 \ \ \text{for all} \ \ p \in \mathcal{P}. \end{aligned} \tag{22}$$

This problem involves a semi-infinite semi-definite constraint on a matrix polynomial in $p$ *only*, i.e. not on the state variables. This allows to construct relaxations that rely on SOS-decomposition for polynomials in $p$, which is the key to keep the size of the resulting LMI-problem *quadratic* in the number of system states, as opposed to the exponential growth of the size of the LMI-problem for the direct approach discussed in Section 3.

## 5 Robust Analysis by SOS-Decompositions

Let us now consider the generic robust LMI problem

$$\begin{array}{l} \text{supremize } \; c^T y \\ \text{subject to } \; F(x,y) \succ 0 \text{ for all } x \in \mathbb{R}^{n_x} \text{ with } g_i(x) \leq 0, \; i = 1, \ldots, n_g \end{array} \tag{23}$$

where $F(x,y) \in \mathcal{S}^r$ and $g_i(x) \in \mathbb{R}$ are polynomial in $x$ and affine in $y$ respectively. The problems (3) and (23) differ in two essential structural properties. First, (3) is just a semi-definite polynomial minimization problem, whereas (23) has a linear objective with a semi-infinite linear SDP constraint, where the dependence on the parameter $x$ is polynomial. Second, in the relaxation for (3) we had to guarantee positivity for *scalar* polynomials, whereas (23) involves a *matrix-valued* positivity constraint. Despite these structural differences the relaxations suggested in this section are similar to those in Section 3.

### 5.1 Scalar Constraints

Let us first consider scalar-valued semi-infinite constraints which corresponds to $F(x,y)$ being of dimension $1 \times 1$ in (23). If for some $y$ there exist $\epsilon > 0$ and SOS polynomials $s_i(x)$, $i = 1, \ldots, n_g$, such that

$$F(x,y) + \sum_{i=1}^{n_g} s_i(x) g_i(x) - \epsilon \;\text{ is SOS in }\; x, \tag{24}$$

then it is very simply to verify that $c^T y$ is a lower bound on the optimal value of (23). The best possible lower bound is achieved with the supremal $c^T y$ over all $\epsilon > 0$ and all SOS polynomials $s_i(x)$ satisfying (24), and we have equality if $G(x) = \text{diag}(g_1(x), \ldots, g_{n_g}(x))$ satisfies the constraint qualification (5). The proof is a a variant of that of Theorem 1 and is hence omitted.

### 5.2 Scalarization of Matrix-Valued Constraints

Let us now assume that $F(x,y)$ is indeed *matrix-valued*. Our intention is to illustrate why a straightforward scalarization technique fails to lead to the desired properties of the corresponding LMI relaxations. Indeed define $f(v,x,y) := v^T F(x,y) v$ and

$$h_i(v,x) = g_i(x) \;\; i = 1, \ldots, n_g, \;\; h_{n_g+1}(v,x) = 1 - v^T v, \;\; h_{n_g+2}(v,x) = v^T v - 2.$$

Then $F(x,y) \succ 0$ for all $x$ with $g_i(x) \leq 0$, $i = 1, \ldots, n_g$, is equivalent to $f(v,x,y) > 0$ for all $(x,v)$ with $h_i(v,x) \leq 0$, $i = 1, \ldots, n_g + 2$. As in Section 5.1 this condition can be relaxed as follows: there exists $\epsilon > 0$ and SOS polynomials $s_i(v,x)$, $i = 1, \ldots, n_g + 2$, such that

$$f(v,x,y) + \sum_{i=1}^{n_g+2} s_i(v,x) h_i(v,x) - \epsilon \;\text{ is SOS}$$

(with exactness of the relaxation under constraint qualifications). Unfortunately, despite $f(v, x, y)$ is quadratic in $v$, no available result allows to guarantee that the SOS polynomials $s_i(v, x)$, $i = 1, \ldots, n_g + 2$, can be chosen quadratic in $v$ without loosing the relaxation's exactness. Hence one has to actually rely on higher order SOS polynomials in $v$ to guarantee that the relaxation gap vanishes. In our specific problem, $v$ has $2n+m_1+1$ components such that the relaxation size grows exponentially in the system state-dimension, and we fail to achieve the desired polynomial growth in $n$.

### 5.3 Matrix-Valued Constraints

This motivates to fully avoid scalarization as follows. We replace (22) by requiring the existence of $\epsilon > 0$ and SOS matrices $S_i(x)$ of the same dimension as $F(x, y)$ such that

$$F(x,y) + \sum_{i=1}^{n_g} S_i(x) g_i(x) - \epsilon I \quad \text{is an SOS matrix in} \quad x. \tag{25}$$

It is easy to see that the optimal value of (23) is bounded from below by the largest achievable $c^T y$ for which there exist $\epsilon > 0$ and SOS matrices $S_i(x)$ with (25). It is less trivial to show that this relaxation is *exact* which has been proved in our paper [31]. This is the key step to see that one can construct a family of LMI relaxations for computing arbitrarily tight lower bounds on the optimal of (23) whose sizes grow exponentially only in the number of components of $x$.

*Remark 6.* We have performed partial dualization (in the high dimensional variables) in order to arrive at the formulation of (23). It is interesting to interpret the replacement of the semi-infinite constraint in (23) by (25) as a second Lagrange relaxation step in the low dimensional variable $x$. In this sense the suggested relaxation can be viewed as a full SOS Lagrange dualization of the original nonlinear semi-definite program, and exponential growth is avoided by the indicated split into two steps.

### 5.4 Verification of Matrix SOS Property

Let us now discuss how to construct a linear SDP representation of (25) if restricting the search of the SOS matrices $S_i(x)$, $i = 1, \ldots, n_g$, to an arbitrary subspace of polynomials matrices. The suggested procedure allows for complete flexibility in the choice of the corresponding monomial basis with a direct and compact description of the resulting linear SDP, even for problems that involve SOS matrices. Moreover it forms the basis for trying to reduce the relaxation sizes for specific problem instances.

For all these purposes let us choose a polynomial vector

$$u(x) = \mathrm{col}(u_1(x), \ldots, u_{n_u}(x))$$

whose components $u_j(x)$ are pairwise different $x$-monomials. Then $S(x)$ of dimension $r \times r$ is said to be SOS with respect to the monomial basis $u(x)$ if there exist real matrices $T_j$, $j = 1, \ldots, n_u$, such that

$$S(x) = T(x)^T T(x) \quad \text{with} \quad T(x) = \sum_{j=1}^{n_u} T_j u_j(x) = \sum_{j=1}^{n_u} T_j (u_j(x) \otimes I_r).$$

If $U = (T_1 \ \cdots \ T_{n_u})$ and if $P$ denotes the permutation that guarantees

$$u(x) \otimes I_r = P[I_r \otimes u(x)],$$

we infer with $W = (UP)^T (UP) \succeq 0$ that

$$S(x) = [I_r \otimes u(x)]^T W [I_r \otimes u(x)]. \tag{26}$$

In order to render this relation more explicit let us continue with the following elementary concepts. If $M \in \mathbb{R}^{nr \times nr}$ is partitioned into $n \times n$ blocks as $(M_{jk})_{j,k=1,\ldots,r}$ define

$$\mathrm{Trace}_r(M) = \begin{pmatrix} \mathrm{Tr}(M_{11}) & \cdots & \mathrm{Tr}(M_{1r}) \\ \vdots & \ddots & \vdots \\ \mathrm{Tr}(M_{r1}) & \cdots & \mathrm{Tr}(M_{rr}) \end{pmatrix}$$

as well as the bilinear mapping $\langle ., . \rangle_r : \mathbb{R}^{mr \times nr} \times \mathbb{R}^{mr \times nr} \to \mathbb{R}^{r \times r}$ as

$$\langle A, B \rangle_r = \mathrm{Trace}_r(A^T B).$$

One then easily verifies that $[I_r \otimes u(x)]^T W [I_r \otimes u(x)] = \langle W, I_r \otimes u(x) u(x)^T \rangle_r$. If we denote the pairwise different monomials in $u(x)u(x)^T$ by $w_j(x)$, $j = 1, \ldots, n_w$, and if we determine the unique symmetric $Q_j$ with

$$u(x)u(x)^T = \sum_{j=1}^{n_w} Q_j w_j(x),$$

we can conclude that

$$S(x) = \sum_{j=1}^{n_w} \langle W, I_r \otimes Q_j \rangle_r \, w_j(x). \tag{27}$$

This proves one direction of the complete characterization of $S(x)$ being SOS with respect to $u(x)$, to be considered as a flexible generalization of the Gram-matrix method [27] to polynomial matrices.

**Lemma 1.** *The matrix polynomial $S(x)$ is SOS with respect to the monomial basis $u(x)$ iff there exist necessarily unique symmetric $S_j$, $j = 1, \ldots, n_w$, such that $S(x) = \sum_{j=1}^{n_w} S_j w_j(x)$, and the linear system*

$$\langle W, I_r \otimes Q_j \rangle_r = S_j, \qquad j = 1, \ldots, n_w, \tag{28}$$

*has a solution $W \succeq 0$.*

**Proof.** If $W \succeq 0$ satisfies (28) we can determine a Cholesky factorization of $PWP^T$ as $U^T U$ to obtain $W = (UP)^T(UP)$ and reverse the above arguments. ■

### 5.5 Construction of LMI Relaxation Families

With monomial vector $v_i(x)$ and some real vector $b_i$ for each $i = 1, 2, \ldots, n_g$ let us represent the constraint functions as $g_i(x) = b_i^T v_i(x), i = 1, 2, \ldots, n_g$. Moreover, let us choose monomial vectors $u_i(x)$ of length $n_i$ to parameterize the SOS matrices $S_i(x)$ with respect to $u_i(x)$ with $W_i \succeq 0$, $i = 0, 1, \ldots, n_g$, as in Section 5.4. With $v_0(x) = 1$ and $b_0 = -1$, we infer

$$\begin{aligned}
-S_0(x) \sum_{i=1}^{n_g} S_i(x) g_i(x) &= \sum_{i=0}^{n_g} \left\langle\, W_i, I_r \otimes \left[u_i(x) u_i(x)^T\right] \,\right\rangle_r \left[b_i^T v_i(x)\right] \\
&= \sum_{i=0}^{n_g} \left\langle\, W_i, \left(I_r \otimes \left[u_i(x) u_i(x)^T\right]\right) \left[b_i^T v_i(x)\right] \,\right\rangle_r \\
&= \sum_{i=0}^{n_g} \left\langle\, W_i, I_r \otimes \left(\left[u_i(x) u_i(x)^T\right] \otimes \left[b_i^T v_i(x)\right]\right) \,\right\rangle_r \\
&= \sum_{i=0}^{n_g} \left\langle\, W_i, I_r \otimes \left(\left[I_{n_i} \otimes b_i^T\right] \left[u_i(x) u_i(x)^T \otimes v_i(x)\right]\right) \,\right\rangle_r .
\end{aligned}$$

Let us now choose the pairwise different monomials

$$w_0(x) = 1, \qquad w_1(x), \ldots, w_{n_w}(x)$$

to allow for the representations

$$u_i(x) u_i(x)^T \otimes v_i(x) = \sum_{j=0}^{n_w} P_{ij} w_j(x) \quad \text{and} \quad F(x, y) = \sum_{j=0}^{n_w} A_j(y) w_j(x), \tag{29}$$

with symmetrically valued $A_j(y)$ that depend affinely on $y$. Then there exist $\epsilon > 0$ and SOS matrices $S_i(x)$ with respect to $u_i(x)$, $i = 0, \ldots, n_g$, such that

$$F(x, y) + \sum_{i=1}^{n_g} S_i(x) g_i(x) - \epsilon I = S_0(x) \tag{30}$$

if and only if there exists a solution to the following LMI system:

$$\epsilon > 0, \quad W_i \succeq 0, \quad i = 0, \ldots, n_g, \tag{31}$$

$$A_0(y) + \sum_{i=0}^{n_g} \left\langle\, W_i, I_r \otimes \left( \left[ I_{n_i} \otimes b_i^T \right] P_{i0} \right) \,\right\rangle_r - \epsilon I = 0, \tag{32}$$

$$A_j(y) + \sum_{i=0}^{n_g} \left\langle W_i, I_r \otimes \left( \left[ I_{n_i} \otimes b_i^T \right] P_{ij} \right) \,\right\rangle_r = 0, \quad j = 1, \ldots, n_w. \tag{33}$$

We can hence easily maximize $c^T y$ over these LMI constraints to determine a lower bound on the optimal value of (23). Moreover these lower bounds are guaranteed to converge to the optimal value of (23) if we choose $u_i(x)$, $i = 0, \ldots, n_g$, to comprise all monomials up to a certain degree, and if we let the degree bound grow to infinity.

### 5.6 Size of the Relaxation of LMI Problem

The size of the LMI relaxation for (23) is easily determined as follows. The constraints are (31), (32) and (33). The condition on the matrices $W_i$, $i = 0, 1, \ldots, n_g$, to be nonnegative definite in (31) comprises for each $i = 0, 1, \ldots, n_g$ one inequality in $\mathcal{S}^{rn_{u_i}}$, where (as mentioned earlier in the text) $r$ and $n_{u_i}$ denote the number of rows in $F(x, y)$ in (23) and the number of monomials for the $i^{\text{th}}$ SOS matrix, $i = 0, 1, \ldots, n_g$, respectively. On top of that (32) adds $r^2$ and (33) adds $n_w r^2$ scalar equation constraints to the LMI problem.

The decision variables in the LMI relaxation are the lower bound $t$, the matrices for the SOS representation $W_i \in \mathcal{S}^{rn_{u_i}}$, $i = 0, 1, \ldots, n_g$, and the original optimization variables $y \in \mathbb{R}^{n_y}$. Since a symmetric matrix in $\mathcal{S}^{rn_{u_i}}$ can be parameterized by a vector in $\mathbb{R}^{\frac{1}{2} rn_{u_i}(rn_{u_i}+1)}$, we end up in total with

$$1 + n_y + \frac{1}{2} \sum_{i=0}^{n_g} rn_{u_i}(rn_{u_i} + 1) \tag{34}$$

scalar variables in our LMI problem.

This number may be further reduced by using an explicit representation of the degrees of freedom in $W_i$ for each $i \in \{0, 1, \ldots, n_g\}$ by constructing a basis for the solution set of the linear constraints (32) and (33). Although this explicit parameterization may lead to a smaller number of variables, we consider for simplicity in the remainder of this text the implicit representation with (32) and (33).

We can further specify the size of the LMI problem for the fixed order $\mathcal{H}_\infty$ synthesis problem in terms of the sizes of dynamical system, which will show that the size of the LMI's depends *quadratically* on the number of states. If $\mathcal{P}$ is a ball of radius $M$ described by (1), then it can be described by one

polynomial inequality such that $n_g = 1$. The number of variables in $y$ is equal to the dimension of the subspace $\mathcal{Z}_N$: $n_y = N = \dim(\mathcal{Z}_N)$. Since $Z$ is a polynomial in $p$ with matrix coefficients in $\mathcal{S}^{n+m_1}$ , $n_y$ grows at most quadratically in $n$. The number $r$ of rows of each sub-block of $F(p, y)$ in (21) is

| Block | $F_{11}(p,z) - t$ | $F_{22}(p,z)$ | $Z(p,z)$ | $1 - \mathrm{Tr}(Z_{22}(p,z))$ |
|---|---|---|---|---|
| Number of rows | 1 | $n$ | $n + m_1$ | 1 |

which results in $r = 2 + 2n + m_1$ rows. This number of rows is reduced by about 50% to $r = 2 + n$ if we replace the utmost right constraint in (15) ($Z(p) \geq 0$ for all $p \in \mathcal{P}$) by requiring $Z(p)$ to be SOS, as we suggested in Remark 5.

The monomial vectors $u_i$, $i = 0, 1$, should be chosen such that $F(\cdot, y)$ can be expressed in terms of $S_0$ and $S_1 g_1$ as in (30) and they will be specified for the concrete examples in Section 6. Note that $n_{u_i}$, $i = 0, 1$, is independent of $n$ since $u_i$, $i = 0, 1$, are polynomials in $p$.

Summarizing, $n_y$ grows at most quadratically in $n$, $r$ grows linearly in $n$ and $n_{u_i}$, $i = 0, 1$, are independent of $n$. Equation (34) therefore implies that the number of LMI variables is indeed *quadratic* in $n$.

*Remark 7.* Although not limiting for unstructured controller synthesis [36], the assumption $D_{22} = 0$ on the direct-feedthrough term is usually restrictive if the controller is required to admit some structure. It is therefore useful to discuss how a nonzero $D_{22}$ can be taken into account in our method. For $D_{22} \neq 0$ the closed-loop matrices are rational functions of $p$:

$$
\begin{aligned}
A(p) &= \begin{pmatrix} A^{\mathrm{ol}} + B_2^{\mathrm{ol}} Q(p) D_K(p) C_2^{\mathrm{ol}} & B_2^{\mathrm{ol}} Q(p) C_K(p) \\ A_{21}(p) & A_K(p) + B_K(p) D_{22} Q(p) C_K(p) \end{pmatrix} \\
B(p) &= \begin{pmatrix} B_1^{\mathrm{ol}} + B_2^{\mathrm{ol}} Q(p) D_K(p) D_{21}^{\mathrm{ol}} \\ B_K(p) D_{21} + B_K(p) D_{22} Q(p) D_K(p) D_{21}^{\mathrm{ol}} \end{pmatrix} \\
C(p) &= \begin{pmatrix} C_1^{\mathrm{ol}} + D_{12} Q(p) D_K(p) C_2^{\mathrm{ol}} & D_{12} Q(p) C_K(p) \end{pmatrix} \\
D(p) &= D_{11}^{\mathrm{ol}} + D_{12}^{\mathrm{ol}} Q(p) D_K(p) D_{21}^{\mathrm{ol}}
\end{aligned}
$$

where $A_{21}(p) := B_K(p) C_2^{\mathrm{ol}} + B_K(p) D_{22} Q(p) D_K(p) C_2^{\mathrm{ol}}$ and

$$
Q(p) := (I - D_K p D_{22})^{-1} .
$$

This results in rational dependence of the left-hand sides of (13) and (14) on $p$, such that we can not directly apply the SOS test. Under the well-posedness condition that $I - D_K(p) D_{22}$ is nonsingular for all $p \in \mathcal{P}$, we can multiply (13) and (14) by $\det(I - D_K(p) D_{22})^2$. Since then $\det(I - D_K(p) D_{22})^2 > 0$ for all $p \in \mathcal{P}$, the resulting inequalities are equivalent to (13) and (14) and

polynomial in $p$. This implies that our problem is a robust polynomial LMI problem (23) and we can apply the SOS technique. It is easy to see that testing well-posedness is a robust LMI problem as well.

*Remark 8.* In a similar fashion we can introduce a rational parameterization for $Z(p)$. Let $Z_j : \mathbb{R}^{n_p} \mapsto \mathcal{S}^{n+m_1}$, $j = 1, 2, \ldots, N$, be a set of linearly independent symmetric valued rational functions in $p$ without poles in $\mathcal{P}$ (instead of polynomials as in Section 4.2). By multiplication of the inequalities (13)-(15) with the smallest common denominator of $Z(p)$ that is strictly positive for all $p \in \mathcal{P}$, their left-hand side becomes a polynomial of $p$. We expect such parameterizations of $Z(p)$ with rational functions to be much better than with polynomials, in the sense that they generate better lower bounds for the same size of the LMI problem. Comparison of these parameterizations is a topic for future research.

## 6 Application

We present lower bound computations for the fixed order $\mathcal{H}_\infty$ problem of a fourth order system and a $27^{\text{th}}$ order active suspension system. The results of this section have also been published in [16].

### 6.1 Fourth Order System

We consider an academic example with

$$\left(\begin{array}{c|c|c} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ \hline C_2 & D_{21} & D_{22} \end{array}\right) = \left(\begin{array}{ccccc|cc} -7 & 4 & 0 & 0.2 & 0.9 & 0.2 & 0 \\ -0.5 & -2 & 0 & 0 & 2 & 0.2 & 0 \\ 3 & 4 & -0.5 & 0 & 0.1 & 0.1 & 0 \\ 3 & 4 & 2 & -1 & -4 & 0 & -0.2 \\ \hline 0 & -10 & -3 & 0 & 0 & 3 & -4 \\ \hline 0.8 & 0.1 & 0 & 0 & 0.3 & 0 & 0 \end{array}\right)$$

and computed lower bounds on the closed-loop $\mathcal{H}_\infty$-performance of all stabilising static controllers in a compact subset of $\mathbb{R}^{2\times 1}$. The open loop $\mathcal{H}_\infty$ norm is 47.6. We first computed an initial feedback law $K_{\text{init}} = \begin{pmatrix} -38 & -28 \end{pmatrix}^T$, which gives a performance of 0.60. We computed bounds for the ball $\mathcal{P}_{\text{ball}} = \{p \in \mathbb{R}^2 |\ \|p\| \le M\}$ with radius $M = 1.5$ around the initial controller, i.e. $K(p) := K_{\text{init}} + \begin{pmatrix} p_1 & p_2 \end{pmatrix}^T$. Observe that $p \in \mathcal{P}$ is equivalent to $g_1(p) \le 0$ where

$$g_1(p) := p_1^2 + p_2^2 - M^2, \tag{35}$$

which is the only constraint on $p$, such that $n_g = 1$ in (23). We therefore optimize over 2 SOS polynomials $S_0$ and $S_1$, both of dimension $10 \times 10$. The choice of the monomials $u_0$ and $u_1$ will be discussed below.

The resulting lower bound, the number of variables in the LMI and the size of the LFT are shown in Table 1 for various degrees in $u_1$ and $Z$. The monomial vector $u_1$ is represented in the table by $(l_1, l_2)$, i.e. $u_1(p) = M^{l_1,l_2}(p)$ where $M$ maps the maximal monomial orders $l_1$ and $l_2$ into a vector containing all mononials

$$p_1^i p_2^j, \qquad 0 \le i \le l_1, \qquad 0 \le j \le l_2.$$

In other words $M^{l_1,l_2}$ is the monomial vector

$$M^{l_1,l_2}(p) := \left( 1 \; p_1 \; p_1^2 \; \dots \; p_1^{l_1} \; p_1 p_2 \; p_1^2 p_2 \; \dots \; p_1^{l_1} p_2 \; \dots \; p_1^{l_1} p_2^{l_2} \right).$$

The vector $(k_1, k_2)$ in the table denotes the maximal monomial degrees for $Z$ in a similar fashion, e.g, $(k_1, k_2) = (2, 1)$ should be read as the parameterization

$$\begin{aligned} Z(p,z) = Z_0 + \textstyle\sum_{j=1}^{N} z_j E_j + z_{j+N} p_1 E_j + z_{j+2N} p_1^2 E_j + z_{j+3N} p_1 p_2 E_j + \\ + \textstyle\sum_{j=1}^{N} z_{j+4N} p_1^2 p_2 E_j + z_{j+5N} p_2 E_j, \end{aligned}$$

where $N = \dim(\mathcal{S}^{n+m_1}) = \frac{1}{2}(n + m_1 + 1)(n + m_1)$ and $E_j$, $j = 1, \dots, N$, is a basis for $\mathcal{S}^{n+m_1}$. For this parameterization the number $n_y$ of variables in $y$ grows quadratically with $n$, which is the maximum growth rate of $n_y$ as mentioned in Section 5.6. The monomial vector $u_0(p) = M^{q_1,q_2}(p)$ is chosen such that the monomials on the right-hand side of (30) match those on the left-hand side, i.e.

$$q_i = \left\lceil \frac{2 + \max\{k_i, 2l_i\}}{2} \right\rceil, \qquad i = 1, 2,$$

where $\lceil x \rceil$ is the smallest integer larger than or equal to $x$. For instance the lower bound in the right lower corner of Table 1 is computed with $q_1 = 3, q_2 = 3$ such that $u_0(p)$ is a monomial vector of length 16.

By a gridding technique we have found an optimal controller $p^{\text{opt}} = \left( 1.33 \; 0.69 \right)^T \in \mathcal{P}$ with performance 0.254. From Table 1 it seems that the lower bound indeed approaches this value for increasing order in both $u_1$ and $Z$. The best lower bound is $t_{\text{opt}} = 0.251$, which is slightly smaller than the optimal performance 0.254. The number of variables in our implementation of the LMI relaxations is shown in Table 2. Each LMI problem has been solved with SeDuMi [34] in at most a few minutes.

### 6.2 Active Suspension System

As a second example we consider the control of an active suspension system, which has been a benchmark system of a special issue of the European

**Table 1.** Lower bounds for $4^{\text{th}}$ order system, various degrees of $Z$ and $S_1$

| $(k_1,k_2)$, monomials in $Z$ | $(l_1,l_2)$, monomials in $u_1$ | | | |
|---|---|---|---|---|
| | $(0,0)$ | $(1,1)$ | $(2,0)$ | $(0,2)$ |
| $(0,0)$ | 0.15584 | 0.16174 | 0.16174 | 0.16174 |
| $(1,0)$ | 0.20001 | 0.20939 | 0.20959 | 0.20183 |
| $(1,1)$ | 0.20319 | 0.21483 | 0.21331 | 0.20785 |
| $(2,0)$ | 0.22259 | 0.2298 | 0.23097 | 0.22396 |
| $(1,2)$ | 0.20642 | 0.22028 | 0.21886 | 0.21171 |
| $(2,1)$ | 0.22669 | 0.23968 | 0.23936 | 0.22959 |
| $(3,0)$ | 0.22361 | 0.24000 | 0.24212 | 0.22465 |
| $(4,0)$ | 0.22465 | 0.24263 | 0.24373 | 0.22504 |
| $(2,2)$ | 0.22737 | 0.24311 | 0.24298 | 0.23277 |
| $(4,2)$ | 0.22889 | 0.25047 | 0.25069 | 0.23414 |

**Table 2.** Number of LMI variables for $4^{\text{th}}$ order system, various degrees of $Z$ and $u_1$

| $(k_1,k_2)$, monomials in $Z$ | $(l_1,l_2)$, monomials in $u_1$ | | | |
|---|---|---|---|---|
| | $(0,0)$ | $(1,1)$ | $(2,0)$ | $(0,2)$ |
| $(0,0)$ | 41 | 1514 | 970 | 1960 |
| $(1,0)$ | 220 | 1528 | 984 | 1974 |
| $(1,1)$ | 413 | 1556 | 1012 | 2002 |
| $(2,0)$ | 234 | 1542 | 998 | 1988 |
| $(1,2)$ | 1211 | 1584 | 1370 | 2030 |
| $(2,1)$ | 881 | 1584 | 1040 | 2030 |
| $(3,0)$ | 578 | 1556 | 1012 | 2002 |
| $(4,0)$ | 867 | 1570 | 1026 | 2016 |
| $(2,2)$ | 1253 | 1626 | 1412 | 2072 |
| $(4,2)$ | 2547 | 2920 | 2706 | 2981 |

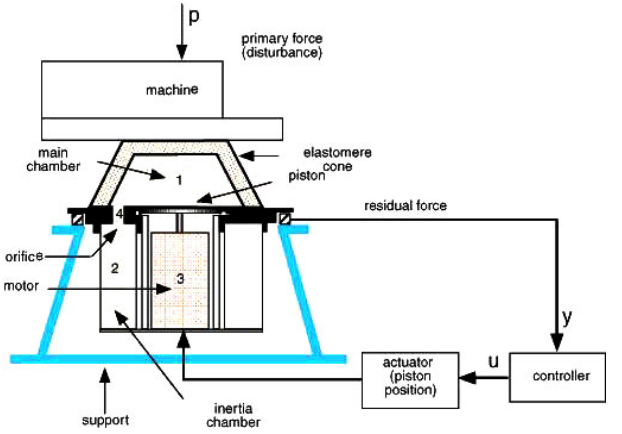



**Fig. 1.** Active suspension system

Journal of Control on fixed-order controller synthesis [21], see Figure 1. The goal is to compute a low-order discrete-time controller such that the closed-loop sensitivity and controller sensitivity satisfy certain frequency-dependent bounds. The system has 17 states and the weights of our 4-block $\mathcal{H}_\infty$ design contributed with 10 states, which add up to 27 states of the generalized plant. The full order design has closed-loop $\mathcal{H}_\infty$-norm 2.48. We computed a $5^{\text{th}}$ order controller by closed-loop balanced residualization with performance 3.41. For more details on the fixed order $\mathcal{H}_\infty$-design the reader is referred to [18]. We computed lower bounds for changes in two diagonal elements of the state-space matrices of the controller

$$K(p) = \left(\begin{array}{c|c} A_K(p) & B_K \\ \hline C_K & D_K \end{array}\right)$$

$$= \left(\begin{array}{cccccc|c} -78.2 & 1129.2 & 173.24 & -97.751 & -130.36 & 6.6086 \\ -1240.9 & -78.2+p_1 & 111.45 & 125.12 & 76.16 & 21.445 \\ 0 & 0 & -6.0294 & 164.81+p_2 & 159 & -11.126 \\ 0 & 0 & 0 & -204.56 & 49.031 & -12.405 \\ 0 & 0 & 0 & -458.3 & -204.56 & -9.4469 \\ \hline -0.067565 & 0.19822 & -1.0047 & -0.069722 & 0.19324 & 0.0062862 \end{array}\right)$$

where $p_1$ and $p_2$ are free scalar controller variables. Table 3 shows computed lower bounds for various degrees in $Z$ and $u_1$ and various controller sets $\mathcal{P}_{\text{ball}} := \{p \quad |\mathbb{R}^{n_p}, \|p\| \leq M\}$, $M \in \{5, 10, 50, 100\}$, together with the number of LMI variables. It is interesting to note that the lower bounds are clearly better than the bounds computed with the S-procedure in [17]. It is a topic

**Table 3.** Lower bounds for suspension system, for various degrees of $Z$ and $S$ and $\mathcal{P}$ balls with various radii.

| monomials | | Radius $M$ of $\mathcal{P} := \{p \mid p \in \mathbb{R}^{n_p}, \|p\| \leq M\}$ | | | | # LMI variables |
|---|---|---|---|---|---|---|
| $Z$ | $u_1$ | 5 | 10 | 50 | 100 | $M \in \{5, 10, 50, 100\}$ |
| $(0,0)$ | $(0,0)$ | 3.2271 | 3.0176 | 2.2445 | 1.9790 | 1719 |
| $(1,0)$ | $(0,0)$ | 3.2570 | 3.0732 | 2.3975 | 2.1585 | 2313 |
| $(0,1)$ | $(0,0)$ | 3.2412 | 3.0468 | 2.3725 | 2.2398 | 2313 |

of our current research to investigate why and for which type of problems the relaxations based on SOS decompositions work better than those based on the S-procedure.

The example illustrates that the lower bound computation is feasible for larger order systems, although the size of the LMI problems still grows gradually.

## 7 Conclusions

We have shown that there exist sequences of SOS relaxations whose optimal value converge from below to the optimal closed-loop $\mathcal{H}_\infty$ performance for controllers of some a priori fixed order. In a first scheme we generalized a well-established SOS relaxation technique for scalar polynomial optimization to problems with matrix-valued semi-definite constraints. In order to avoid the resulting exponential growth of the relaxation size in both the number of controller parameters and system states, we proposed a second technique based on two sequential Lagrange dualizations. This translated the synthesis problem into a problem as known from robustness analysis for systems affected by time-varying uncertainties. We suggested a novel relaxation scheme based on SOS matrices that is guaranteed to be asymptotically exact, and that allows to show that the size of the relaxations only grow quadratically in the dimension of the system state.

We have applied the method to systems of McMillan degree 4 and 27 respectively. The first example illustrated that the lower bounds indeed converge to the optimal fixed-order $\mathcal{H}_\infty$-performance value. The second example showed the feasibility of the approach for plants with moderate McMillan degree, in the sense that we can compute nontrivial lower bounds by solving LMI problems with about 2300 variables.

**Acknowledgements.**
The second author would like to thank Didier Henrion, Pablo Parrilo, Andrea Garulli, and Michael Overton for stimulating discussions.

## A Proof of Strict Feasibility of the Dual Problem

Let us prove that (12) is strictly feasible for all $p_0 \in \mathcal{P}$. For an arbitrary $p_0 \in \mathcal{P}$, we need to show that there exists some $W$ with

$$A(p_0)W_{11} + W_{11}A(p_0)^T + B(p_0)W_{12}^T + W_{12}B(p_0)^T \succ 0, \qquad (36)$$
$$\mathrm{Tr}(W_{22}) < 1, \qquad W \succ 0.$$

Since $(A(p_0), B(p_0))$ is controllable, one can construct an anti-stabilizing state-feedback gain, i.e., a matrix $K$ such that $A(p_0) + B(p_0)K$ has all its eigenvalues in the open right-half plane. Hence there exists some $P \succ 0$ with

$$(A(p_0) + B(p_0)K)P + P(A(p_0) + B(p_0)K)^T \succ 0 \qquad (37)$$

and $rP$ also satisfies (37) for any $r > 0$. Then $W$ defined by the blocks $W_{11} = rP$, $W_{12}^T = rKP$ and

$$W_{22} = W_{12}^T W_{11}^{-1} W_{12} + rI = r(K^T PK + I)$$

in the partition (11) satisfies (36) and $W \succ 0$ for arbitrary $r > 0$. The constructed $W$ does the job if we choose in addition $r > 0$ sufficiently small to achieve $\mathrm{Tr}(W_{22}) = r\mathrm{Tr}(K^T PK + I) < 1$ .

## References

1. Apkarian P, Noll D (2002) Fixed-order $\mathcal{H}_\infty$ control design via an augmented lagrangian method. Technical report, CERT-ONERA, Toulouse, France
2. Beran E, Grigoriadis K (1997) Proceedings of the American Control Conference, Albuquerque, New Mexico:81–85
3. Boyd S, Vandenberghe L (1997) Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In: Communications, Computation, Control and Signal Processing: A Tribute to Thomas Kailath. Kluwer Boston
4. Chesi G, Garulli A, Tesi A, Vicino A (2000) Proceedings of the. $39^{\mathrm{th}}$ IEEE Conference on Decision and Control, Sydney, Australia:1501–1506
5. Chesi G, Garulli A, Tesi A, Vicino A (2002) Homogeneous Lyapunov functions for systems with structured uncertainties, preprint
6. Chilali M, Gahinet P (1996) IEEE Transactions on Automatic Control, 41:358–367
7. Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) IEEE Transactions on Automatic Control 34:831–846

8. Geromel JC, de Souza CC, Skelton RE (1994) Proceedings of the American Control Conference, Baltimore, Maryland:40–44
9. El Ghaoui L, Oustry F, AitRami M (1997) IEEE Transactions on Automatic Control 42:1171–1176
10. Golub GH, Van Loan CF (1989) Matrix Computations. John Hopkins, Baltimore
11. Grassi E, Tsakalis KS (1996) Proceedings of the 35$^{\text{th}}$ IEEE Conference on Decision and Control, Kobe Japan:4776–4781
12. Grassi E, Tsakalis KS, Dash S, Gaikwad SV, MacArthur W, Stein G (2001) IEEE Transactions on Control Systems Technology 9:285–294
13. Grimble MJ, Johnson MA (1999) Proceedings of the American Control Conference, San Diego, California:4368–4372
14. Henrion D(2003) Proceedings of the IEEE Conference on Decision and Control, Maui, Hawaii:4646–4651
15. Henrion D, Sebek M, and Kucera V (2003) IEEE Transactions on Automatic Control 48:1178–1186
16. Hol CWJ, Scherer CW, (2004) Submitted to Conference on Decision and Control, Bahamas.
17. Hol CWJ, Scherer CW, (2004) Proceedings of the American Control Conference, San Diego, California
18. Hol CWJ, Scherer CW, Van der Meché EG, Bosgra OH (2003) European Journal of Control 9: 13–28
19. Ibaraki S, Tomizuka M (2001) Transactions of the ASME, the Journal of Dynamic Systems, Measurement and Control 123:544–549
20. Iwasaki T, Skelton RE (1995) International Journal of Control, 62:1257–1272
21. Landau ID, Karimi A, Miskovic K, Prochazka H (2003) European Journal of Control 9:3–12
22. Lasserre JB (2001) SIAM Journal of Optimization 11:796–817
23. Leibfritz F, Mostafa ME (2002) SIAM Journal on Optimization 12:1048–1074
24. Kojima M (2003) Sums of squares relaxations of polynomial semidefinite programs. Technical report, Tokyo Institute of Technolgy.
25. Parrilo PA (2000) Structured Semidefinite Prograns and Semialgebraic Geometry Methods in Robustness and Optimization. PhD thesis, California Institute of Technology
26. Pillai H, Willems JC (2002) SIAM Journal on Control and Optimization 40:1406–1430
27. Powers V, Wörmann T (1998) Journal on Pure and Applied Algebra 127:99–104
28. Putinar M (1993) Indiana Univiversity Mathematical Journal 42:969–984
29. Rudin W (1976) Principles of Mathematical Analysis. McGraw-Hill, London
30. Scherer CW (2003) Proceedings of the IEEE Conference on Decision and Control, Maui, Hawaii:4652–4657
31. Scherer CW Hol CWJ (2004) Proceedings of the Sixteenth International Symposium on Mathematical Theory of Networks and Systems (MTNS), Leuven
32. Schmüdgen K (1991) Math. Ann. 289:203–206
33. Schweighofer M (2003) Optimization of polynomials on compact semialgebraic sets. Preprint
34. Sturm JF(1999) Optimization Methods and Software 11-12:625–653
35. Tuan HD, Apkarian P (2000) IEEE Transactions on Automatic Control 45: 2111–2117

36. Zhou K, Doyle JC, Glover K (1995) Robust and Optimal Control Prentice Hall, New Jersey

# LMI Optimization for Fixed-Order $H_\infty$ Controller Design

Didier Henrion[1,2]

[1] LAAS-CNRS, 7 Avenue du Colonel Roche, 31 077 Toulouse, France.
henrion@laas.fr
[2] Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Prague, Czech Republic.

A general $H_\infty$ controller design technique is proposed for scalar linear systems, based on properties of positive polynomial matrices. The order of the controller is fixed from the outset, independently of the order of the plant and weighting functions. A sufficient LMI condition is used to overcome non-convexity of the original design problem. The key design step, as well as the whole degrees of freedom are in the choice of a central polynomial, or desired closed-loop characteristic polynomial.

## 1 Introduction

This paper is a continuation of our research work initiated in [8], where a linear matrix inequality (LMI) method was described to design a fixed-order controller robustly stabilizing a linear system affected by polytopic structured uncertainty. A convex LMI approximation of the stability domain in the space of coefficients of a polynomial was obtained there, based on recent results on positive polynomials and strictly positive real (SPR) functions. As explained in [8], the key ingredient in the design procedure resides in the choice of a central polynomial, or desired nominal closed-loop characteristic polynomial.

Paper [8] focused only on polynomial polytope stabilization, which we believe is an interesting research problem, but remains obviously very far from an actual engineering design problem. So in this paper we try to build upon the ideas of [8] to derive a more practical controller design methodology. Standard design specifications are formulated in the frequency domain, on peak values of Bode magnitude plots of (possibly weighted) system transfer functions, this is the so-called $H_\infty$ optimization framework surveyed e.g. in [11].

The main characteristics of our approach, and its contribution with respect to existing work in the area are as follows:

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 73–85, 2005.

- The order of the controller is fixed from the outset, which overcomes the standard limitation of state-space $H_\infty$ techniques that the order of the controller must be at least the same as the order of the plant. Note that, as a consequence, with our techniques using weighting functions does not entail increasing the controller order;
- We use convex optimization over positive polynomials and SPR rational functions, just as in [4, 16, 3]. The main distinction is that we do not use the infinite dimensional Youla-Kučera parametrization of all stabilizing controllers as in [4, 16, 3], or analytical (stable) rational functions in $H_\infty$ [6], so that it is not necessary to resort to model reduction techniques to derive a low-order controller;
- As in our previous work [8], all the degrees of freedom in the design procedure are captured in the choice of the so-called central polynomial, or desired closed-loop characteristic polynomial. Note however that, contrary to the design procedure of [13] or [18], the central polynomial will not necessarily be the actual characteristic polynomial, but only a reference polynomial around which the design is carried out. Influence of the central polynomial on closed-loop performance is generally easy to predict. A general rule of thumb is that open-loop stable poles must be mirrored in the central polynomial, completed by sufficiently fast additional dynamics. This is in contrast with the recent work in [14], where fixed order $H_\infty$ controller design is carried out with the help of Nevanlinna-Pick interpolation, but the influence of design parameters (the so-called spectral zeros) cannot be easily characterized.

Complementary features of our approach are as follows:

- We can enforce LMI structural constraints on the controller coefficients. For example, we can enforce the controller to be strictly proper, or a PID. We can also minimize the Euclidean norm of controller coefficients if suitable;
- Contrary to standard $H_\infty$ techniques, there are no assumptions on open-loop dynamics, presence of zeros along the imaginary axis, properness of weighting functions etc.;
- Continuous-time and discrete-time systems are treated in a unified way, as well as pole location in arbitrary half-plane or disks;

Finally, here is a list of current limitations of our $H_\infty$ design technique:

- As in [8], we use a sufficient convex (LMI) conditions that ensure closed-loop specifications, possibly at the price of some conservatism. We are not aware of any reliable method for measuring the amount of conservatism of our method, even though the many numerical examples we have treated seem to indicate that the approach generally performs at least as well as other design techniques;
- Contrary to well-established state-space $H_\infty$ techniques, the numerical behavior and performance of LMI or semidefinite programming solvers

on these optimization problems over polynomials is still unclear. In the conclusion we mention some research directions and some recent references focusing on this important issue;

- Similarly to standard $H_\infty$ techniques, our design technique is iterative, and a trial-and-error approach cannot be avoided to choose appropriately the central polynomial.

## 2 Problem Statement

The scalar $H_\infty$ design problem to be solved in this paper can be formally stated as follows, based on Kučera's algebraic polynomial formulation [10].

**Problem 1.** Given a set of polynomials $n_i^k(s)$, $d_i^k(s)$ for $i = 1, 2, \ldots$, $k = 1, 2, \ldots$, as well as a set of positive real numbers $\gamma^k$, seek polynomials $x_i(s)$ of given degrees such that

$$\left\| \frac{\sum_i n_i^k(s) x_i(s)}{\sum_i d_i^k(s) x_i(s)} \right\|_\infty < \gamma^k, \quad k = 1, 2, \ldots \tag{1}$$

In the above inequalities

$$\|S\|_\infty = \sup_{s \in \partial\mathcal{D}} |S(s)|$$

denotes the peak value of the magnitude of rational transfer function $S$ when evaluated along the one-dimensional boundary $\partial\mathcal{D}$ of a given stability region

$$\mathcal{D} = \{ s \in \mathbb{C} : \begin{bmatrix} 1 \\ s \end{bmatrix}^\star \begin{bmatrix} d_{11} & d_{12} \\ d_{12}^\star & d_{22} \end{bmatrix} \begin{bmatrix} 1 \\ s \end{bmatrix} < 0 \}$$

of the complex plane, where the star denotes transpose conjugate and Hermitian matrix

$$D = \begin{bmatrix} d_{11} & d_{12} \\ d_{12}^\star & d_{22} \end{bmatrix}$$

has one strictly positive eigenvalue and one strictly negative eigenvalue. Standard choices for $\mathcal{D}$ are the left half-plane ($d_{11} = 0, d_{12} = 1, d_{22} = 0$) and the unit disk ($d_{11} = -1, d_{12} = 0, d_{22} = 1$). Other choices of scalars $d_{11}$, $d_{12}$ and $d_{22}$ correspond to arbitrary half-planes and disks.

The above $H_\infty$ design paradigm covers all the standard frequency domain specifications arising in scalar control problems. For example, in the feedback system of figure 1 the sensitivity of the control system output $z$ to disturbances $v$ is characterized by the sensitivity function

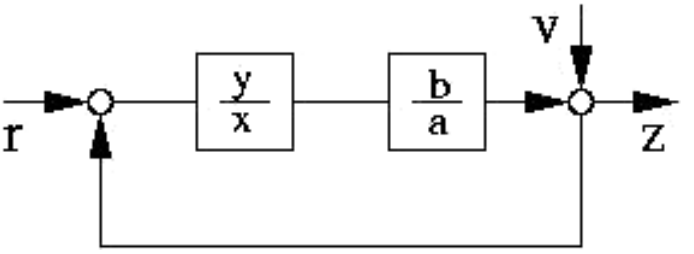



**Fig. 1.** Standard feedback configuration.

$$S = \frac{1}{1 + \frac{b}{a}\frac{y}{x}} = \frac{ax}{ax + by}$$

where plant polynomials $a$ and $b$ are given, and controller polynomials $x$ and $y$ must be found, see [11]. As shown in [4], robustness of the closed-loop plant to model uncertainty may be characterized by the complementary sensitivity function

$$T = 1 - S = \frac{by}{ax + by}$$

which is also the closed-loop system transfer function. As recalled in [2], simplified yet sensible design specifications for a control law can be formulated as

$$\|S\|_\infty < \gamma_S, \quad \|T\|_\infty < \gamma_T$$

where typical values of $\gamma_S$ range between 1.2 and 2.0 and typical values of $\gamma_T$ range between 1.0 and 1.5. This $H_\infty$ control problem, as well as many others, can be formulated using the general paradigm proposed above.

## 3 $H_\infty$ Design Technique

Dropping for convenience the dependence on polynomial indeterminate $s$ and index $k$, the $H_\infty$ design inequality of problem 1 on polynomials $x_i$

$$\left\| \frac{\sum_i n_i x_i}{\sum_i d_i x_i} \right\|_\infty < \gamma$$

can be written equivalently as

$$\mathrm{Re} \left( \frac{\gamma + \frac{\sum_i n_i x_i}{\sum_i d_i x_i}}{\gamma - \frac{\sum_i n_i x_i}{\sum_i d_i x_i}} \right) = \mathrm{Re} \left( \frac{\gamma(\sum_i d_i x_i) + \sum_i n_i x_i}{\gamma(\sum_i d_i x_i) - \sum_i n_i x_i} \right) > 0 \qquad (2)$$

where Re denotes the real part of a complex number. In the above inequalities it is implicit that polynomial indeterminate $s$ describes the stability boundary,

so that all the polynomials are frequency-dependent complex numbers when $s \in \partial\mathcal{D}$.

In order to simplify notations, define

$$n = \gamma(\sum_i d_i x_i) + \sum_i n_i x_i, \quad d = \gamma(\sum_i d_i x_i) - \sum_i n_i x_i \tag{3}$$

and notice that strict positive realness requirement (2)

$$\mathrm{Re}\,\frac{n}{d} = \frac{1}{2}\left(\frac{n}{d} + \frac{n^\star}{d^\star}\right) = \frac{\mathrm{Re}\,n\,\mathrm{Re}\,d + \mathrm{Im}\,n\,\mathrm{Im}\,d}{\|d\|_2^2} > 0$$

is equivalent to the geometric argument condition

$$\cos(n,d) = \frac{\mathrm{Re}\,n\,\mathrm{Re}\,d + \mathrm{Im}\,n\,\mathrm{Im}\,d}{\|n\|_2\|d\|_2} > 0$$

or

$$|(n,d)| < \frac{\pi}{2} \tag{4}$$

where $(n,d)$ denotes the angle between complex numbers $n$ and $d$.

Now introduce an auxiliary polynomial $c$, referred to as the central polynomial for reasons that should become clear later on.

**Lemma 1.** *Geometric condition (4) is equivalent to the existence of a central polynomial $c$ such that*

$$|(n,c)| < \frac{\pi}{4}, \quad |(d,c)| < \frac{\pi}{4}$$

*or equivalently*

$$2\cos^2(n,c) > 1, \quad 2\cos^2(d,c) > 1. \tag{5}$$

**Proof:** Suppose first that polynomial $c$ exists such that inequalities (5) are satisfied. Geometrically, it follows that the angle between $n$ and $d$ never exceeds $\pi/2$, which is inequality (4). Conversely, just choose $c = n + d$ as a valid central polynomial, and then inequalities (5) hold. □

**Lemma 2.** *Inequalities (5) hold if and only if*

$$\begin{bmatrix} \mathrm{Re}\,n^\star c & \mathrm{Im}\,n^\star c \\ \mathrm{Im}\,n^\star c & \mathrm{Re}\,n^\star c \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \mathrm{Re}\,d^\star c & \mathrm{Im}\,d^\star c \\ \mathrm{Im}\,d^\star c & \mathrm{Re}\,d^\star c \end{bmatrix} \succeq 0 \tag{6}$$

*where $\succeq 0$ means positive semidefinite.*

**Proof:** The first inequality in (5) can be written explicitly as

$$2(\mathrm{Re}\, n\, \mathrm{Re}\, c + \mathrm{Im}\, n\, \mathrm{Im}\, c)^2 > (\mathrm{Re}^2 n + \mathrm{Im}^2 n)(\mathrm{Re}^2 c + \mathrm{Im}^2 c)$$

or equivalently

$$(\mathrm{Re}\, n\, \mathrm{Re}\, c + \mathrm{Im}\, n\, \mathrm{Im}\, c)^2 > (\mathrm{Re}\, n\, \mathrm{Im}\, c - \mathrm{Im}\, n\, \mathrm{Re}\, c)^2.$$

Using a Schur complement argument, this can be reformulated as a 2-by-2 positive semidefiniteness constraint

$$\begin{bmatrix} \mathrm{Re}\, n\, \mathrm{Re}\, c + \mathrm{Im}\, n\, \mathrm{Im}\, c & \mathrm{Re}\, n\, \mathrm{Im}\, c - \mathrm{Im}\, n\, \mathrm{Re}\, c \\ \mathrm{Re}\, n\, \mathrm{Im}\, c - \mathrm{Im}\, n\, \mathrm{Re}\, c & \mathrm{Re}\, n\, \mathrm{Re}\, c + \mathrm{Im}\, n\, \mathrm{Im}\, c \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\, n^\star c & \mathrm{Im}\, n^\star c \\ \mathrm{Im}\, n^\star c & \mathrm{Re}\, n^\star c \end{bmatrix} \succeq 0.$$

The second matrix inequality in (6) is obtained similarly. □

Defining the 2-by-2 polynomial matrices

$$N(s) = \begin{bmatrix} n(s) & 0 \\ 0 & n(s) \end{bmatrix}, \quad D(s) = \begin{bmatrix} d(s) & 0 \\ 0 & d(s) \end{bmatrix}, \quad C(s) = \begin{bmatrix} c(s) & c(s) \\ -c(s) & c(s) \end{bmatrix}$$

inequalities (6) can also be written as

$$N^\star(s)C(s) + C^\star(s)N(s) \succeq 0, \quad D^\star(s)C(s) + C^\star(s)D(s) \succeq 0 \tag{7}$$

for $s \in \partial\mathcal{D}$. Inequalities (7) are positivity conditions on polynomial matrices .

Controller parameters, i.e. coefficients of polynomials $x_i(s)$, enter linearly in polynomials $n(s)$ and $d(s)$, as well as in polynomial matrices $N(s)$ and $D(s)$. So it means that as soon as central polynomial $c$ is given, positivity conditions (7) are linear in design parameters. Positivity conditions on polynomial matrices depending linearly on design parameters can be formulated as LMIs as follows.

Let

$$N = \begin{bmatrix} N_0 & N_1 & \cdots & N_\delta \end{bmatrix}, \quad D = \begin{bmatrix} D_0 & D_1 & \cdots & D_\delta \end{bmatrix}, \quad C = \begin{bmatrix} C_0 & C_1 & \cdots & C_\delta \end{bmatrix}$$

denote matrix coefficients of powers of indeterminate $s$ in polynomial matrices $N(s)$, $D(s)$ and $C(s)$ respectively, where $\delta$ is the highest degree arising in polynomials $n(s)$, $d(s)$ and $c(s)$. Define

$$\Pi = \begin{bmatrix} I_2 & & & 0 \\ & \ddots & & \vdots \\ & & I_2 & 0 \\ 0 & I_2 & & \\ \vdots & & \ddots & \\ 0 & & & I_2 \end{bmatrix}$$

as a matrix of size $4\delta$-by-$2(\delta+1)$, together with the linear mapping

$$H(P) = \Pi^T (H \otimes P)\Pi = \Pi^T \begin{bmatrix} aP & bP \\ b^\star P & cP \end{bmatrix} \Pi$$

where square matrix $P$ has dimension $2\delta$ and $\otimes$ denotes the Kronecker product. Then the following result is a corollary of lemma 2 in [7].

**Lemma 3.** *Given polynomial matrix $C(s)$, polynomial matrices $N(s)$ and $D(s)$ satisfy positivity conditions (7) if and only if there exist matrices $P_n = P_n^\star$ and $P_d = P_d^\star$ such that*

$$N^\star C + C^\star N - H(P_n) \succ 0, \quad D^\star C + C^\star D - H(P_d) \succ 0. \tag{8}$$

Repeating this argument on all the frequency domain specifications (1), we obtain the following central result:

**Theorem 1.** *Given polynomials $n_i^k(s)$, $d_i^k(s)$, positive scalars $\gamma^k$ and central polynomials $c^k(s)$ for $k = 1, 2, \ldots$, there exist polynomials $x_i(s)$ solving $H_\infty$ design problem 1 if problem (8) is feasible for each index $k$. This is a convex LMI problem in coefficients of polynomials $x_i(s)$.*

Based on the discussion of [8], central polynomial $c$ plays the role of a target closed-loop characteristic polynomial around which the design is carried out. In particular, setting the degree of central polynomial $c$ also sets the degree of polynomials $n$ and $d$ in (3), as well as the degree of controller polynomials $x_i$. Sensible strategies for the choice of central polynomial $c$ are discussed in [8], but a general rule of thumb is that open-loop stable poles must be mirrored in the central polynomial, completed by sufficiently fast additional dynamics.

Note finally that, since LMI (8) is a sufficient condition to enforce $H_\infty$ specifications (1), generally these specifications will be satisfied with a certain amount of conservatism, i.e. $\gamma^k$ is always an upper bound on the actual $H_\infty$ norm in (1) achieved by feedback.

## 4 Numerical Examples

The $H_\infty$ design technique of theorem 1 has been implemented in a documented Matlab 6.5 m-file available at

`www.laas.fr/~henrion/software/hinfdes`

that will be included to the next release 3.0 of the commercial Polynomial Toolbox [15] for Matlab. An alternate code that does not require the Polynomial Toolbox can be obtained by contacting the author.

The numerical examples were treated with the help of Matlab 6.5 running under SunOS release 5.8 on a SunBlade 100 workstation. Operations on polynomials were performed with the Polynomial Toolbox 2.5 [15]. The LMI problems were solved with SeDuMi 1.05 [17] with default tuning parameters, interfaced with SeDuMi Interface [12].

### 4.1 Optimal Robust Stability

Consider the optimal robust stability problem of section 11.1 in [4], where the open-loop plant in figure 1 is given by

$$\frac{b}{a} = \frac{s-1}{(s+1)(s-0.5)}$$

and we seek a controller $y/x$ minimizing $\gamma_T$ under the following weighted $H_\infty$ constraint on the closed-loop transfer function

$$\|WT\|_\infty = \left\| \left( \frac{s+0.1}{s+1} \right) \left( \frac{by}{ax+by} \right) \right\|_\infty < \gamma_T.$$

The following Matlab code seeks a first order controller for $\gamma_T = 1.9$:

```
a = (s+1)*(s-0.5); b = (s-1); gammaT = 1.9;
c = (s+0.1)*(s+1)^2*(s+3); % central polynomial
lmi = hinfdes([],'init',[1 1]); % seek first order controller
lmi = hinfdes(lmi,(s+0.1)*[0 b],(s+1)*[a b],c,gammaT); % H-inf spec
out = hinfdes(lmi,'solve'); % solve LMI
x = out(1); y = out(2);
```

Central polynomial $c$ is the key design parameter, and together with upper bound $\gamma_T$ they capture the whole degrees of freedom. Roots in $c$ are just an indication on where closed-loop poles should be located: generally, roots of characteristic polynomial $ax + by$ will be located around roots of $c$, but they may also differ significantly due to structural constraints. The $H_\infty$ design procedure then consists in iteratively playing with the roots of $c$, while lowering upper bound $\gamma_T$.

In table 1 we show different choices of roots for $c$, denoted by $\sigma(c)$ (4 roots = 2 for the open-loop system, 1 for the weighting function, 1 for the controller), together with actual poles of closed-loop transfer function $T$ (3 roots) denoted by $\sigma(ax+by)$, upper bounds $\gamma_T$ and the actual weighted norms $\|WT\|_\infty$ achieved by the computed controllers. Each design requires about 1 second of CPU time on our computer.

We can see that a good strategy is to start with a central polynomial with all its roots in $-1$, and a loose upper bound on $\gamma_T$. Decreasing $\gamma_T$, some closed-loop poles move away from $-1$, which gives indications on how to move roots of the central polynomial. At the bottom of the table, we can see that by allowing a very fast root in the central polynomial, $\gamma_T$ can

**Table 1.** Optimal robust stability. Roots of central polynomial, characteristic polynomial, $H_\infty$ upper bound and achieved $H_\infty$-norm.

| $\sigma(c)$ | $\sigma(ax+by)$ | $\gamma_T$ | $\|WT\|_\infty$ |
|---|---|---|---|
| -1,-1,-1,-1 | $-1.04 \pm i1.08$, -0.230 | 2.9 | 2.11 |
| -1,-1,-1,-0.1 | $-0.731 \pm i0.566$, -0.118 | 2.3 | 1.74 |
| -2,-1,-1,-0.1 | $-1.133 \pm i0.586$, -0.114 | 2.1 | 1.54 |
| **-3,-1,-1,-0.1** | $\mathbf{-1.383 \pm i0.642}$, **-0.0932** | **1.9** | **1.47** |
| -10,-1,-1,-0.1 | -6.775, -1.063, -0.1059 | 1.8 | 1.31 |
| -500,-1,-1,-0.1 | -1700, -0.992, -0.103 | 1.7 | 1.21 |

be decreased significantly close to the theoretical infimum of 1.20. Yet the closed-loop system also features a very fast pole, and the resulting controller $y/x = (-2046.2 - 2039.7s)/(3744.0 + s)$ results impractical.

A good tradeoff here is indicated in boldface letters in table 1, where a weighted $H_\infty$-norm of 1.47 is achieved with the first-order controller

$$\frac{y}{x} = \frac{-3.0456 - 3.2992s}{5.6580 + s}.$$

Note however that the sensitivity function has very poor norm $\|S = 1-T\|_\infty = 13.1$, due to the fact that no specifications were enforced on $S$. As a result, the above controller can be very sensitive to perturbations, or fragile, as pointed out in [9].

A more sensible design approach would then enforce an additional $H_\infty$ specification on $S$, such as

$$\|S\|_\infty = \left\| \frac{ax}{ax+by} \right\|_\infty < \gamma_S$$

for some suitable value of $\gamma_S$. However, as shown in [1], for this numerical example the ratio between the unstable open-loop pole and zero is small so there is no controller that will give a reasonably robust closed-loop system.

Adding a line to the above Matlab code to enforce an additional specification on $\|S\|_\infty$, we obtain (after about 2 seconds of CPU time) with $c(s) = (s+1)^3(s+100)$, $\gamma_T = 4$ and $\gamma_S = 4$ the following first-order controller

$$\frac{y}{x} = \frac{-873.30 - 816.37s}{1202.4 + s}$$

producing $\|S\|_\infty = 3.44$ and $\|WT\|_\infty = 2.24$.

### 4.2 Flexible Beam

Consider the flexible beam example of section 10.3 in [4]. The open-loop plant in figure 1 is given by

$$\frac{b}{a} = \frac{-6.4750s^2 + 4.0302s + 175.7700}{5s^4 + 3.5682s^3 + 139.5021s^2 + 0.0929s}.$$

For the closed-loop plant to approximate a standard second-order system with settling time at 1% of 8 seconds and overshoot less then 10%, the following frequency domain specification on the weighted sensitivity function is enforced in [4]:

$$\|WS\|_\infty = \left\| \left( \frac{s^2 + 1.2s + 1}{s(s + 1.2)} \right) \left( \frac{ax}{ax + by} \right) \right\|_\infty < \gamma_S.$$

Suppose we are looking for a second-order controller. The open-loop plant has poles 0, $-0.6660 \cdot 10^{-3}$, and $-0.3565 \pm i5.270$, and the weighting function has poles at 0 and $-1.2$. The central polynomial must mirror open-loop stable poles, so an initial choice of central polynomial features roots $-0.6660 \cdot 10^{-3}$, $-0.3565 \pm i5.270$, $-1.2$ plus two roots at $-10^{-2}$ corresponding to the open-loop plant integrator and the weighting function integrator, plus two roots at $-1$ (arbitrary) corresponding to the controller poles. With this choice of central polynomial and $\gamma_S = 5$ the $H_\infty$ LMI problem is solved in 15 seconds but the resulting step response is too slow.

After a series of attempts, an acceptable step response was obtained with the roots $\sigma(c) = \{-0.6660 \cdot 10^{-3}, -10^{-2}, -0.3565 \pm i5.270, -0.1, -1, -1, -1\}$ corresponding to the central polynomial $c(s) = 0.1858 \cdot 10^{-4} + 0.3000 \cdot 10^{-1}s + 3.178s^2 + 37.33s^3 + 94.06s^4 + 90.50s^5 + 33.45s^6 + 3.824s^7 + s^8$. With $\gamma_S = 5$ function `hinfdes` returns the controller

$$\frac{y}{x} = \frac{0.77489 \cdot 10^{-4} + 0.16572 \cdot 10^{-1}s + 0.36537s^2}{0.41025 \cdot 10^{-1} + 1.0437s + s^2}$$

producing

$$\|S\|_\infty = 1.27, \quad \|T\|_\infty = 1.01$$

and a step response with settling time at 1% of 11.3 seconds and overshoot of 4%. Bode magnitude plots of $S$ and $T$ are given in figure 2, and the step response is shown in figure 3. Note that a similar performance was obtained in [4] with a controller of eighth order.

## 5 Conclusion

We have proposed an iterative $H_\infty$ design technique where all the degrees of freedom are on the choice of a central polynomial, or desired closed-loop characteristic polynomial around which the design is carried out. Contrary to most of the existing $H_\infty$ optimization techniques, the order of the controller is fixed from the very outset, independently of the order of the plant or weighting functions. Our $H_\infty$ design method is based on results on positive polynomial matrices and convex optimization over LMIs . As a result, it could be easily implemented in a Matlab and SeDuMi framework.

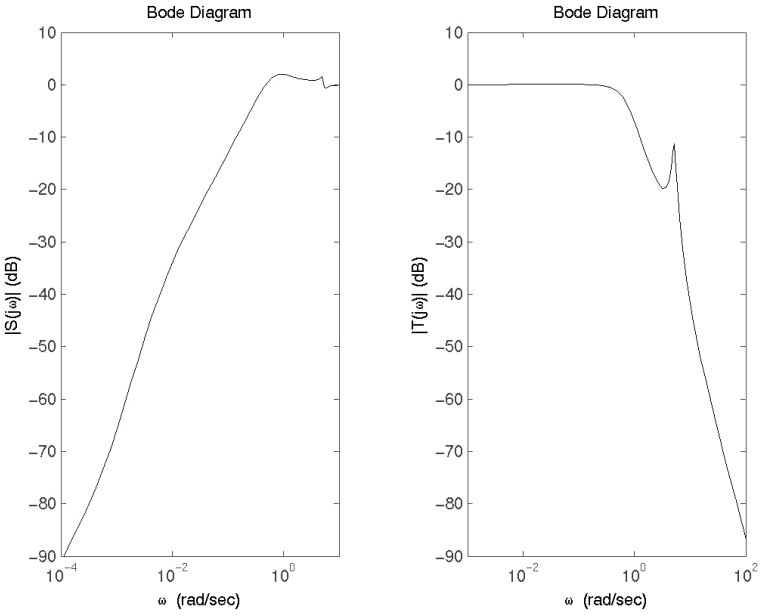



**Fig. 2.** Flexible beam. Bode magnitude plots of $S$ and $T$.

We believe that a promising research direction may be the study of numerical properties (computational complexity, numerical stability) of algorithms tailored to solve the structured LMI problems arising from the theory of positive polynomials and polynomial matrices. As shown in [5], the Hankel or Toeplitz structure can be exploited to design fast algorithms to solve Newton steps in barrier schemes and interior-point algorithms. Numerical stability is also a concern, since it is well-known for example that Hankel matrices are exponentially ill-conditioned. Alternative polynomial bases such as Chebyshev or Bernstein polynomials may prove useful.

**Acknowledgements.**
Support of the Grant Agency of the Czech Republic under Project No. 102/02/0709 is acknowledged. This paper benefited from stimulating discussions with Zdeněk Hurák and Dimitri Peaucelle.

## References

1. Åström K J (2000). Limitations on Control System Performance. European Journal of Control, 6:2–20.

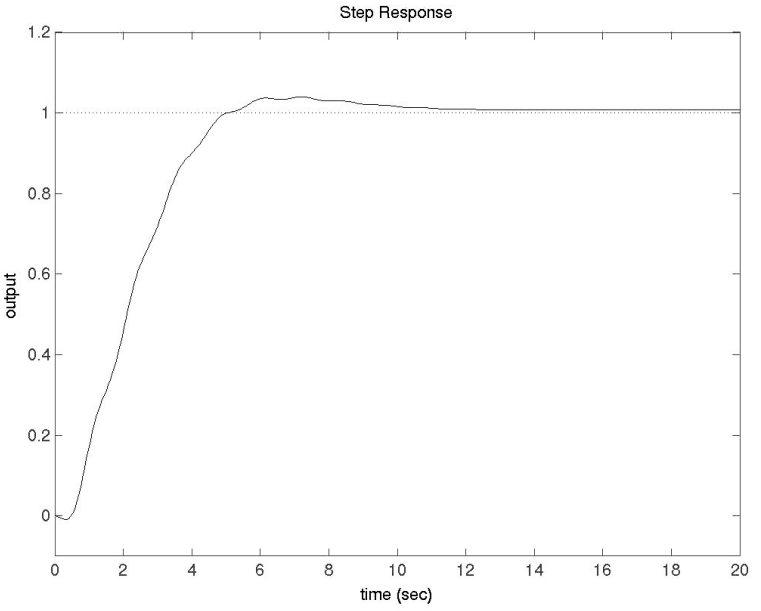



**Fig. 3.** Flexible beam. Step response.

2. Åström K J, Panagopoulos H, Hägglund T (1998). Design of PI controllers based on non-convex optimization. Automatica, 34(5):585–601.
3. Dorato P (2000). Analytic Feedback System Design: An Interpolation Approach. Brooks Cole Publishing, New York.
4. Doyle J C, Francis B A, Tannenbaum A R (1992). Feedback Control Theory. MacMillan, New York.
5. Genin Y, Hachez Y, Nesterov Yu, Ştefan R, Van Dooren P, Xu S (2002). Positivity and LMIs. European Journal of Control, 8:275–298.
6. Helton J W, Merino O (1998). Classical control using $H_\infty$ methods. SIAM, Philadelphia.
7. Henrion D, Arzelier D, Peaucelle D (2003). Positive polynomial matrices and improved LMI robustness conditions. Automatica, 39(8):1479–1485.
8. Henrion D, Šebek M, Kučera V (2003). Positive polynomials and robust stabilization with fixed-order controllers. IEEE Transactions on Automatic Control, 48(7):1178-1186.
9. Keel L H, Bhattacharyya S P (1997). Robust, fragile or optimal ? IEEE Transactions on Automatic Control, 42(8):1098–1105.
10. Kučera V (1979). Discrete Linear Control: The Polynomial Approach. John Wiley and Sons, Chichester.
11. Kwakernaak H (1993). Robust control and $H_\infty$ optimization – Tutorial Paper. Automatica, 29(2):255–273.

12. Labit Y, Peaucelle D, Henrion D (2002). SeDuMi Interface 1.02 : a Tool for Solving LMI Problems with SeDuMi. Proceedings of the IEEE Conference on Computer-Aided Control System Design, Glasgow.
13. Langer J, Landau I D (1999). Combined Pole Placement/Sensitivity Function Shaping Method using Convex Optimization Criteria. Automatica, 35(6):1111–1120.
14. Nagamune R (2002). Robust control with complexity constraint: a Nevanlinna-Pick interpolation approach. PhD Thesis, Royal Institute of Technology, Stockholm, Sweden.
15. PolyX, Ltd (2000). The Polynomial Toolbox for Matlab. Version 2.5. Prague, Czech Republic.
16. Rantzer A, Megretski A (1994). A Convex Parameterization of Robustly Stabilizing Controllers. IEEE Transactions on Automatic Control, 39(9):1802–1808.
17. Sturm J F (1999). Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones. Optimization Methods and Software, 11-12:625–653.
18. Wang S, Chow J H (2000). Low-Order Controller Design for SISO Systems Using Coprime Factors and LMI. IEEE Transactions on Automatic Control, 45(6):1166–1169.

# An LMI-Based Technique for Robust Stability Analysis of Linear Systems with Polynomial Parametric Uncertainties

Graziano Chesi[1], Andrea Garulli[1], Alberto Tesi[2], and Antonio Vicino[1]

[1] Dipartimento di Ingegneria dell'Informazione, Università di Siena,
{chesi,garulli,vicino}@dii.unisi.it
[2] Dipartimento di Sistemi e Informatica, Università di Firenze,
tesi@dsi.unifi.it

Robust stability analysis of state space models with respect to real parametric uncertainty is a widely studied challenging problem. In this paper, a quite general uncertainty model is considered, which allows one to consider polynomial nonlinearities in the uncertain parameters. A class of parameter-dependent Lyapunov functions is used to establish stability of a matrix depending polynomially on a vector of parameters constrained in a polytope. Such class, denoted as Homogeneous Polynomially Parameter-Dependent Quadratic Lyapunov Functions (HPD-QLFs), contains quadratic Lyapunov functions whose dependence on the parameters is expressed as a polynomial homogeneous form. Its use is motivated by the property that the considered matricial uncertainty set is stable if and only there exists a HPD-QLF. The paper shows that a sufficient condition for the existence of a HPD-QLF can be derived in terms of Linear Matrix Inequalities (LMIs).

## 1 Introduction

Establishing robust stability of a system affected by parametric uncertainty is a challenging problem that has been addressed since long time [1, 2]. Typical state space uncertainty models considered in the literature include *interval matrices* and *polytopes of matrices*. While satisfactory results have been obtained for special classes of interval matrices (see [3, 4] and references therein), the problem of assessing robust stability of generic polytopes of matrices is computationally hard.

A standard way to tackle the problem is to formulate sufficient conditions, based on the existence of a common Lyapunov function for all the matrices in the polytope. Common quadratic Lyapunov functions (see e.g. [5]) provide a viable solution, because their existence can be checked via the solution

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 87–101, 2005.

of a system of Linear Matrix Inequalities (LMIs), that are special convex optimizations [6]. On the other hand, it is well known that the existence of a common Lyapunov function is strictly related to the circle criterion of absolute stability theory [7], and therefore the resulting sufficient condition can be quite conservative.

In recent years, several techniques involving parameter-dependent quadratic Lyapunov functions have been proposed in order to reduce conservatism [8, 9, 10, 11]. Most of these techniques employ quadratic Lyapunov functions that depend linearly on the uncertain parameters. In [12], a new class of parameter-dependent quadratic Lyapunov functions has been introduced, for which the dependence on the uncertain parameters is expressed as a polynomial homogeneous form. Such class, denoted as Homogeneous Polynomially Parameter-Dependent Quadratic Lyapunov Functions (simply abbreviated as HPD-QLFs), has been shown to provide less conservative sufficient condition for the stability of a polytope of matrices, with respect to linearly parameter-dependent Lyapunov functions.

In this paper, a more general uncertainty model is considered, which allows one to consider polynomial nonlinearities in the uncertain parameters. The aim of the paper is to show that HPD-QLFs can be used to study robust stability of matrices depending polynomially on uncertain parameters constrained in a polytope. The main motivation for using this class of Lyapunov function, is that it is rich enough to provide a complete answer to the robust stability problem. Specifically, it turns out that the considered matricial uncertainty model is stable if and only if there exists a HPD-QLF.

It is worth remarking that the potential of homogeneous polynomial forms for the analysis of control systems has been recognized since long time (see e.g., [13, 14]). In recent years, homogeneous forms gained a renewed interest, motivated by the strong connection with semidefinite programming and convex optimization techniques [15], which made it possible to exploit them for solving several problems (see e.g. [16]).

The paper is organized as follows. The robust stability problem is formulated in Section 2. A complete parameterization of homogeneous matricial forms is provided in Section 3: this is essential in order to formulate sufficient conditions for positivity of such forms in terms of LMIs. The main contribution of the paper, i.e. the sufficient condition to determine the sought HPD-QLF, is given in Section 4. The case of continuous-time systems is treated in detail, while the extension to the discrete-time case is briefly sketched. Numerical examples are provided in Section 5 and concluding remarks are given in Section 6.

## 2 Problem Formulation and Preliminaries

The following notation is adopted in the paper:

- $0_n, 0_{m\times n}$: origin of $\mathbb{R}^n$ and of $\mathbb{R}^{m\times n}$;

- $\mathbb{R}^n_0$: $\mathbb{R}^n \setminus \{0_n\}$;
- $I_n$: identity matrix $n \times n$;
- $A'$: transpose of matrix $A$;
- $A > 0$ ($A \geq 0$): symmetric positive definite (semidefinite) matrix $A$;
- $A \otimes B$: Kronecker's product of matrices $A$ and $B$;
- $\mathrm{sv}([p_1, p_2, \ldots, p_q]') \triangleq [p_1^2, p_2^2, \ldots, p_q^2]'$.

Consider the continuous-time state space model

$$\dot{x}(t) = A(p)x(t), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, and $p = [p_1, p_2, \ldots, p_q]' \in \mathbb{R}^q$ is the uncertain parameter vector which belongs to the set

$$\mathcal{P} = \left\{ p \in \mathbb{R}^q : \sum_{i=1}^{q} p_i = 1, \quad p_i \geq 0, \ i = 1, 2, \ldots, q \right\}. \tag{2}$$

The matrix $A(p)$ is assumed homogeneous in $p$ of degree $r$, that is

$$A(p) = \sum_{\substack{i_1 \geq 0, \ldots, i_q \geq 0 \\ i_1 + \ldots + i_q = r}} p_1^{i_1} \cdots p_q^{i_q} \, A_{i_1 \ldots i_q} \tag{3}$$

where $A_{i_1 \ldots i_q} \in \mathbb{R}^{n \times n}$ are given real matrices. Consider the set of matrices defined as

$$\mathcal{A} = \left\{ A(p) \in \mathbb{R}^{n \times n} : \ p \in \mathcal{P} \right\}. \tag{4}$$

The problem we address can be stated as follows.

**Robust stability problem**:
Establish if the set $\mathcal{A}$ in (4) is Hurwitz, i.e., $\mathcal{A}$ contains only Hurwitz matrices.

It is worth observing that the uncertainty model (4) is fairly general, as it encompasses also families of matrices $A(p)$ whose dependence on $p$ is polynomial (not necessarily homogeneous). Indeed, for any polynomial matrix

$$\tilde{A}(p) = \sum_{\substack{i_1 \geq 0, \ldots, i_q \geq 0 \\ i_1 + \ldots + i_q \leq r}} p_1^{i_1} \cdots p_q^{i_q} \, \tilde{A}_{i_1 \ldots i_q} \tag{5}$$

let us define the homogeneous polynomial matrix $A(p)$ such that

$$A(p) = \sum_{\substack{i_1 \geq 0, \ldots, i_q \geq 0 \\ i_1 + \ldots + i_q \leq r}} \left( \sum_{i=1}^{q} p_i \right)^{r - i_1 - \ldots - i_q} p_1^{i_1} \cdots p_q^{i_q} \, \tilde{A}_{i_1 \ldots i_q}. \tag{6}$$

It turns out that $A(p) = \tilde{A}(p)$ for all $p \in \mathcal{P}$: therefore, there is no loss of generality in considering $A(p)$ as a homogeneous matricial form in $p$.

The key step for addressing the robust stability problem formulated above is the construction of a Homogeneous Polynomially Parameter-Dependent Quadratic Lyapunov Function (simply abbreviated as HPD-QLF)

$$v_m(x;p) = x' P_m(p) x, \tag{7}$$

where $P_m(p) \in \mathbb{R}^{n \times n}$ is a *homogeneous matricial form* of degree $m$, i.e., a matrix whose entries are (real $q$-variate) homogeneous forms of degree $m$.

Let us introduce the following important property of HPD-QLFs.

**Lemma 1.** *The set $\mathcal{A}$ is Hurwitz if and only if there exists a HPD-QLF $v_m(x;p)$ such that*

$$\begin{cases} P_m(p) > 0 \\ A'(p)P_m(p) + P_m(p)A(p) < 0 \end{cases} \quad \forall p \in \mathcal{P}. \tag{8}$$

Proof. Sufficiency is obvious. Regarding the necessity, let us suppose that $\mathcal{A}$ is Hurwitz. Let $E_{\bar{m}}(p) = E'_{\bar{m}}(p)$ be any homogeneous matricial form of degree $\bar{m}$ such that $E_{\bar{m}}(p) > 0 \ \forall p \in \mathcal{P}$, and let us consider the Lyapunov equation

$$A'(p)P(p) + P(p)A(p) = -E_{\bar{m}}(p). \tag{9}$$

The solution is a rational matricial function $P(p) = P'(p) > 0 \ \forall p \in \mathcal{P}$, whose entries have homogeneous numerators of degree $m$ and the same denominator $d(p)$, such that $d(p) > 0 \ \forall p$. Hence, one can write $P(p)$ as $P(p) = d^{-1}(p)P_m(p)$. Then, it clearly follows that $P_m(p)$ satisfies (8) (in particular, $A'(p)P_m(p) + P_m(p)A(p) = -d(p)E_{\bar{m}}(p)$). ■

**Remark 1**. A result analogous to that of Lemma 1 can be easily obtained for discrete-time systems.

**Remark 2**. From the proof of Lemma 1, an upper bound on the degree $m$ of the homogeneous matricial form $P_m(p)$ defining the HPD-QLF can be derived. In particular, by choosing $E_{\bar{m}}(p) = E_0$ constant, one has $m < \frac{1}{2}rn(n+1)$.

## 3 Parameterization of Homogeneous Matricial Forms

In order to give sufficient conditions for the existence of a HPD-QLF, it is useful to introduce a suitable parameterization of homogeneous matricial forms.

First, let us recall the Complete Square Matricial Representation (CSMR) of homogeneous scalar forms, which provides all possible representations of a homogeneous polynomial form of degree $2m$ in terms of a quadratic form in

the space of the monomials of degree $m$ (see [17] for details). Let $w_{2m}(p)$ be a homogeneous form of degree $2m$ in $p \in \mathbb{R}^q$. The CSMR of $w_{2m}(p)$ is defined as

$$w_{2m}(p) = p^{\{m\}'}(W_m + L_m(\alpha))p^{\{m\}}$$

where:

- $p^{\{m\}} \in \mathbb{R}^{\sigma(q,m)}$ is the vector containing all monomials of degree $m$ in $p$;
- $W_m \in \mathbb{R}^{\sigma(q,m)\times\sigma(q,m)}$ is a suitable symmetric matrix;
- $\alpha \in \mathbb{R}^{\sigma_{par}(q,m)}$ is a vector of free parameters;
- $L_m(\alpha)$ is a linear parameterization of the set $\mathcal{L}_m = \left\{L_m = L_m' : \; p^{\{m\}'}L_m p^{\{m\}} = 0 \quad \forall p \in \mathbb{R}^q\right\}$.

The quantities $\sigma(q,m)$ and $\sigma_{par}(q,m)$ are given respectively by ([17])

$$\sigma(q,m) = \frac{(q+m-1)!}{(q-1)!m!}, \tag{10}$$

$$\sigma_{par}(q,m) = \frac{1}{2}\sigma(q,m)[\sigma(q,m)+1] - \sigma(q,2m). \tag{11}$$

Similarly to what has been done for scalar forms, one can introduce the CSMR for homogeneous matricial forms. Let $C_{2m}(p) \in \mathbb{R}^{n\times n}$ be a homogeneous matricial form of degree $2m$ in $p \in \mathbb{R}^q$. Then, $C_{2m}(p)$ can be written as

$$C_{2m}(p) = \left(p^{\{m\}} \otimes I_n\right)' \bar{C}_m \left(p^{\{m\}} \otimes I_n\right) \tag{12}$$

where $\bar{C}_m \in \mathbb{R}^{n\sigma(q,m)\times n\sigma(q,m)}$ is a suitable matrix (denoted hereafter as a SMR matrix of $C_{2m}(p)$). Such a matrix is not unique and, indeed, all the matrices $\bar{C}_m$ describing $C_{2m}(p)$ are given by

$$\bar{C}_m + \bar{U}_m, \quad \bar{U}_m \in \mathcal{U}_m \tag{13}$$

where

$$\begin{aligned}\mathcal{U}_m = \Big\{&\bar{U}_m = \bar{U}_m' \in \mathbb{R}^{n\sigma(q,m)\times n\sigma(q,m)} : \\ &\left(p^{\{m\}} \otimes I_n\right)' \bar{U}_m \left(p^{\{m\}} \otimes I_n\right) = 0_{n\times n} \quad \forall p \in \mathbb{R}^q\Big\}.\end{aligned} \tag{14}$$

**Lemma 2.** *The set $\mathcal{U}_m$ in (14) is a linear space of dimension*

$$u(q,n,m) = \frac{1}{2}n\Big\{\sigma(q,m)[n\sigma(q,m)+1] - (n+1)\sigma(q,2m)\Big\}. \tag{15}$$

Proof. Set $\mathcal{U}_m$ is a linear space since $\bar{Z}_1, \bar{Z}_2 \in \mathcal{U}_m \Rightarrow z_1\bar{Z}_1 + z_2\bar{Z}_2 \in \mathcal{U}_m \; \forall z_1, z_2 \in \mathbb{R}$. Now, let us observe that $n\sigma(q,m)(n\sigma(q,m)+1)/2$ is the number of entries of a symmetric matrix of dimension $n\sigma(q,m) \times n\sigma(q,m)$, while $n(n+1)\sigma(q,2m)/2$ is the number of independent terms (and, hence, of

constraints) of a homogeneous $n \times n$ matricial form of degree $2m$ in $q$ variables. ■

Let $\bar{U}_m(\alpha)$, $\alpha \in \mathbb{R}^{u(q,n,m)}$, be a linear parameterization of $\mathcal{U}_m$. The CSMR of $C_{2m}(p)$ is hence given by

$$C_{2m}(p) = \left(p^{\{m\}} \otimes I_n\right)' \left(\bar{C}_m + \bar{U}_m(\alpha)\right) \left(p^{\{m\}} \otimes I_n\right). \tag{16}$$

## 4 Robust Stability Analysis via HPD-QLFs

In this section, sufficient conditions for robust stability are provided in terms of LMIs. The aim is to find a HPD-QLF as in (7), such that $P_m(p)$ satisfies (8) in Lemma 1. The first condition to be satisfied is the positive definiteness of the HPD-QLF matrix $P_m(p)$ within the set $\mathcal{P}$, i.e. the first inequality in (8). In this respect, a parameterization of positive definite matrices $P_m(p)$ is provided next, in Section 4.1. The second inequality in (8) will be dealt with in Section 4.2.

### 4.1 Parameterization of Positive Definite HPD-QLF Matrices

The following result exploits a basic property of homogeneous forms to give an alternative characterization of the positivity of $P_m(p)$.

**Lemma 3.** *The condition*

$$P_m(p) > 0 \quad \forall p \in \mathcal{P} \tag{17}$$

*holds if and only if*

$$P_m(\mathrm{sv}(p)) > 0 \quad \forall p \in \mathbb{R}^q_0. \tag{18}$$

Proof. Being $P_m(p)$ homogeneous in $p$, one has that (17) is equivalent to $P_m(\kappa p) > 0$ for all $p \in \mathcal{P}$ and for all positive $\kappa$, and hence to $P_m(p) > 0$ for all $p$ in the positive orthant (i.e., such that $p_i \geq 0$, $i = 1, \ldots, q$, and $p \neq 0_n$). The latter condition can be equivalently expressed as in (18). ■

**Remark 3**. Notice that Lemma 3 still holds if the parametric uncertainty region $\mathcal{P}$ is replaced by any set of the form $\{p : \ p_i \geq 0, i = 1, \ldots, q; \ \|p\|_\ell = \gamma\}$, for any norm $\|\cdot\|_\ell$ and positive $\gamma$.

Observe that $P_m(\mathrm{sv}(p))$ can be written as

$$P_m(\mathrm{sv}(p)) = \left(p^{\{m\}} \otimes I_n\right)' \bar{S}_m \left(p^{\{m\}} \otimes I_n\right) \tag{19}$$

for some suitable matrix $\bar{S}_m \in \mathcal{S}_m$ where

$$\mathcal{S}_m = \Big\{ \bar{S}_m = \bar{S}_m' \in \mathbb{R}^{n\sigma(q,m)\times n\sigma(q,m)} : \ \left(p^{\{m\}} \otimes I_n\right)' \bar{S}_m \left(p^{\{m\}} \otimes I_n\right) \text{ does not contain entries } p_1^{i_1} p_2^{i_2} \dots p_q^{i_q} \text{ with any odd } i_j \Big\}. \tag{20}$$

From the definition of $\mathcal{S}_m$, an alternative way to write (19) is

$$P_m(\mathrm{sv}(p)) = \tilde{T}_m \left( [\mathrm{sv}(p)]^{\{m\}} \otimes I_n \right) \tag{21}$$

where $\tilde{T}_m \in \mathbb{R}^{n\times n\sigma(q,m)}$ is a suitable matrix. Hence, due to Lemma 3, one has that if $\bar{S}_m$ in (19) is positive definite, then the matrix

$$P_m(p) = \tilde{T}_m \left( p^{\{m\}} \otimes I_n \right)$$

is positive definite for $p \in \mathcal{P}$.
In order to increase the degrees of freedom in the selection of $P_m(p)$, it is worth noticing that matrix $\bar{S}_m$ in (19) is not unique. The next lemma provides a characterization of the set $\mathcal{S}_m$.

**Lemma 4.** *The set $\mathcal{S}_m$ is a linear space of dimension*

$$s(q,n,m) = \frac{1}{2} n \Big\{ \sigma(q,m)[n\sigma(q,m)+1] - (n+1)[\sigma(q,2m) - \sigma(q,m)] \Big\}. \tag{22}$$

Proof. The set $\mathcal{S}_m$ is a linear space since $\bar{Z}_1, \bar{Z}_2 \in \mathcal{S}_m \Rightarrow z_1\bar{Z}_1 + z_2\bar{Z}_2 \in \mathcal{S}_m \ \forall z_1, z_2 \in \mathbb{R}$. Now, let us observe that $n\sigma(q,m)(n\sigma(q,m)+1)/2$ is the number of entries of a symmetric matrix of dimension $n\sigma(q,m) \times n\sigma(q,m)$, while $n(n+1)(\sigma(q,2m) - \sigma(q,m))/2$ is the number of independent terms (and, hence, of constraints) containing at least one odd power of a homogeneous $n \times n$ matricial form of degree $2m$ in $q$ variables. ■

Let $\bar{S}_m(\beta)$, $\beta \in \mathbb{R}^{s(q,n,m)}$, be a linear parameterization of $\mathcal{S}_m$. Clearly, this induces a corresponding linear parameterization $\tilde{T}_m(\beta)$ of matrix $\tilde{T}_m$ in (21). Hence, one can choose the family of candidate HPD-QLF matrices

$$P_m(p;\beta) = \tilde{T}_m(\beta) \left( p^{\{m\}} \otimes I_n \right) \tag{23}$$

which depends linearly on the parameterization $\beta$ of $\mathcal{S}_m$. Following the above reasoning, one has the next result, which is the key step for the formulation of the sufficient condition for solving the robust stability problem.

**Lemma 5.** *Let $\bar{S}_m(\beta)$ belong to $\mathcal{S}_m$ in (20). If $\bar{S}_m(\beta) > 0$, then*

$$P_m(p;\beta) > 0 \quad \forall p \in \mathcal{P}.$$

### 4.2 LMI-Based Sufficient Conditions for Robust Stability

In the following, a sufficient condition for the solution of the robust stability problem is provided. To this purpose, let us introduce the homogeneous matricial form of degree $m + r$

$$Q_{m+r}(p;\beta) = -A'(p)P_m(p;\beta) - P_m(p;\beta)A(p) \tag{24}$$

and the related homogeneous form $Q_{m+r}(sv(p);\beta)$, which can be written as

$$Q_{m+r}(\mathrm{sv}(p);\beta) = \left(p^{\{m+r\}} \otimes I_n\right)' \bar{R}_{m+r}(\beta) \left(p^{\{m+r\}} \otimes I_n\right), \tag{25}$$

where $\bar{R}_{m+r}(\beta) \in \mathbb{R}^{n\sigma(q,m+r)\times n\sigma(q,m+r)}$ is any SMR matrix of $Q_{m+r}(\mathrm{sv}(p);\beta)$. The following result yields the sought sufficient condition.

**Theorem 1.** *The set $\mathcal{A}$ in (4) is Hurwitz if there exist a nonnegative integer $m$, and parameter vectors $\alpha \in \mathbb{R}^{u(q,n,m+r)}$ and $\beta \in \mathbb{R}^{s(q,n,m)}$ such that*

$$\begin{cases} \bar{S}_m(\beta) > 0 \\ \bar{R}_{m+r}(\beta) + \bar{U}_{m+r}(\alpha) > 0 \end{cases} \tag{26}$$

*where $\bar{S}_m(\beta) \in \mathcal{S}_m$, $\bar{U}_{m+r}(\alpha) \in \mathcal{U}_{m+r}$, and $\bar{R}_{m+r}(\beta)$ is defined by (24)-(25).*

Proof. First, let $P_m(p;\beta)$ be defined as in (23). Then, from (26) and Lemma 5 one has that $P_m(p;\beta) > 0\ \forall p \in \mathcal{P}$, and hence the first condition in (8) holds. Second, let us observe that $\bar{R}_{m+r}(\beta)+\bar{U}_{m+r}(\alpha)$ is the CSMR matrix of $Q_{m+r}(\mathrm{sv}(p);\beta)$ in (25). Hence, (26) implies that $Q_{m+r}(\mathrm{sv}(p);\beta) > 0\ \forall p \in \mathbb{R}^q_0$. From Lemma 3 it turns out that $Q_{m+r}(p;\beta) > 0\ \forall p \in \mathcal{P}$ and, therefore, $\mathcal{A}$ is Hurwitz. ■

The inequalities (26) form an LMI feasibility problem with $s(q,n,m) + u(q,n,m+r)$ free parameters. The size of the matrices is $n\sigma(q,m)$ for the first inequality and $n\sigma(q,m+r)$ for the second one. The solution can be computed by using efficient convex optimization tools, like [18, 19].

A question that naturally arises is whether there exists a relationship between the families of HPD-QLFs of degree $m$ and $m+1$. The following result clarifies that, if the sufficient condition of Theorem 1 is satisfied for $m$, then it is satisfied also for $m+1$.

**Theorem 2.** *Let $m$ be a nonnegative integer. If there exist parameter vectors $\alpha \in \mathbb{R}^{u(q,n,m+r)}$ and $\beta \in \mathbb{R}^{s(q,n,m)}$ such that (26) is satisfied, then there exist parameter vectors $\tilde{\alpha} \in \mathbb{R}^{u(q,n,m+r+1)}$ and $\tilde{\beta} \in \mathbb{R}^{s(q,n,m+1)}$ such that*

$$\begin{cases} \bar{S}_{m+1}(\tilde{\beta}) > 0 \\ \bar{R}_{m+r+1}(\tilde{\beta}) + \bar{U}_{m+r+1}(\tilde{\alpha}) > 0 \end{cases}. \tag{27}$$

Proof. From the proof of Theorem 1 we have that, $\forall p \in \mathcal{P}$, $P_m(p;\beta) > 0$ and $Q_{m+r}(p;\beta) > 0$. Define now $P_{m+1}(p) = P_m(p;\beta)\sum_{i=1}^{q} p_i$. It follows that $P_{m+1}(p) > 0\ \forall p \in \mathcal{P}$ and $Q_{m+r+1}(p) = Q_{m+r}(p;\beta)\sum_{i=1}^{q} p_i > 0\ \forall p \in \mathcal{P}$. This means that $v_{m+1}(x;p) = x'P_{m+1}(p)x$ is a HPD-QLF of degree $m+1$ satisfying the condition of Lemma 1.
Let us show now that $P_{m+1}(\mathrm{sv}(p))$ admits a positive definite SMR matrix, that is there exists $\tilde{\beta}$ such that $\bar{S}_{m+1}(\tilde{\beta}) > 0$. Let $K_{m+1}$ be the matrix satisfying

$$p \otimes p^{\{m\}} = K_{m+1}p^{\{m+1\}}\ \forall p \in \mathbb{R}^q.$$

Then,

$$\begin{aligned}
P_{m+1}(\mathrm{sv}(p)) &= \left(\sum_{i=1}^{q} p_i^2\right)\left(p^{\{m\}} \otimes I_n\right)' \bar{S}_m(\beta)\left(p^{\{m\}} \otimes I_n\right) \\
&= p'p\left(p^{\{m\}} \otimes I_n\right)' \bar{S}_m(\beta)\left(p^{\{m\}} \otimes I_n\right) \\
&= \left(p \otimes p^{\{m\}} \otimes I_n\right)' \left(I_q \otimes \bar{S}_m(\beta)\right)\left(p \otimes p^{\{m\}} \otimes I_n\right) \\
&= \left(K_{m+1}p^{\{m+1\}} \otimes I_n\right)' \left(I_q \otimes \bar{S}_m(\beta)\right)\left(K_{m+1}p^{\{m+1\}} \otimes I_n\right) \\
&= \left(p^{\{m+1\}} \otimes I_n\right)' (K_{m+1} \otimes I_n)' \left(I_q \otimes \bar{S}_m(\beta)\right)(K_{m+1} \otimes I_n) \\
&\quad \left(p^{\{m+1\}} \otimes I_n\right) \\
&= \left(p^{\{m+1\}} \otimes I_n\right)' \bar{S}_{m+1}\left(p^{\{m+1\}} \otimes I_n\right)
\end{aligned} \tag{28}$$

where

$$\bar{S}_{m+1} = (K_{m+1} \otimes I_n)' \left(I_q \otimes \bar{S}_m(\beta)\right)(K_{m+1} \otimes I_n).$$

From (28), it is clear that $\bar{S}_{m+1} \in \mathcal{S}_{m+1}$, and hence there exists $\tilde{\beta}$ such that $\bar{S}_{m+1}(\tilde{\beta}) = \bar{S}_{m+1}$. Moreover, since $\bar{S}_m(\beta) > 0$ and $K_{m+1}$ is a matrix with full column rank, it follows that $\bar{S}_{m+1}(\tilde{\beta}) > 0$.
Let us show now that $Q_{m+r+1}(\mathrm{sv}(p))$ admits a positive definite SMR matrix. Following the same development as in (28), one gets

$$Q_{m+r+1}(\mathrm{sv}(p)) = \left(p^{\{m+r+1\}} \otimes I_n\right)' \bar{R}_{m+r+1}(\tilde{\beta})\left(p^{\{m+r+1\}} \otimes I_n\right) \tag{29}$$

where

$$\bar{R}_{m+r+1}(\tilde{\beta}) = (K_{m+r+1} \otimes I_n)' \left(I_q \otimes \left(\bar{R}_{m+r}(\beta) + \bar{U}_{m+r}(\alpha)\right)\right)(K_{m+r+1} \otimes I_n). \tag{30}$$

Since $\bar{R}_{m+r}(\beta) + \bar{U}_{m+r}(\alpha) > 0$ it follows that $\bar{R}_{m+r+1}(\tilde{\beta}) > 0$. Therefore, $Q_{m+r+1}(\mathrm{sv}(p))$ admits the positive definite SMR matrix $\bar{R}_{m+r+1}(\tilde{\beta}) + \bar{U}_{m+r+1}(\tilde{\alpha})$ with $\tilde{\alpha} = 0_{u(q,n,m+r+1)}$, and (27) holds. ■

**Remark 4**. The proposed technique can be applied also to discrete-time systems $x(t+1) = A(p)x(t)$, where $A(p)$ belongs to the set in (4). An LMI-based sufficient condition similar to (26) can be obtained by observing that

$$P_m(p) - A'(p)P_m(p)A(p) = P_m(p)\left(\sum_{i=1}^{q} p_i\right)^{2r} - A'(p)P_m(p)A(p)$$

and the right term is a homogeneous matricial form of degree $m+2r$ that can be parameterized as

$$Q_{m+2r}(p;\beta) = P_m(p;\beta)\left(\sum_{i=1}^{q} p_i\right)^{2r} - A'(p)P_m(p;\beta)A(p).$$

Then, the sought sufficient condition is obtained by following the same reasoning that has led to Theorem 1. A result analogous to Theorem 2 can also be derived.

## 5 Numerical Examples

In this section, some numerical examples are presented to illustrate the proposed technique for robust stability analysis of uncertain systems.

### 5.1 Example 1

The first example is deliberately simple, in order to show how the LMIs involved in the sufficient condition are generated. Consider the problem of computing the robust parametric margin $\rho^*$ defined as

$$\rho^* = \sup\left\{\rho \in \mathbb{R} : \ \hat{A}(\theta) \text{ is Hurwitz for all } \theta \in [0,\rho]\right\}$$

where

$$\hat{A}(\theta) = \begin{bmatrix} -1 & -1 \\ 4 & -1 \end{bmatrix} + \theta \begin{bmatrix} 0 & -7 \\ -13 & 3 \end{bmatrix} + \theta^2 \begin{bmatrix} 0 & 6 \\ 14 & -2 \end{bmatrix}.$$

The equivalent matrix $A(p;\rho)$ is computed by setting $\theta = \rho\, p_1$ and converting the so-obtained polynomial matrix in $p_1$ in a homogeneous one with respect to $p = [p_1, p_2]'$ as described in (5)–(6) where $p_2 = 1 - p_1$. In particular, we have

$$A(p;\rho) = p_1^2 \begin{bmatrix} -1 & -1 - 7\rho + 6\rho^2 \\ 4 - 13\rho + 14\rho^2 & -1 + 3\rho - 2\rho^2 \end{bmatrix}$$
$$+ p_1 p_2 \begin{bmatrix} -2 & -2 - 7\rho \\ 8 - 13\rho & -2 + 3\rho \end{bmatrix} + p_2^2 \begin{bmatrix} -1 & -1 \\ 4 & -1 \end{bmatrix}$$

with $q = n = r = 2$. Note that the solution of the above problem amounts to solving a one-parameter family of robust stability problems addressed in the paper, namely one for each fixed value of $\rho$.

Consider first the case $m = 0$, which means that a common Lyapunov function is sought for all the matrices of the polytope $\mathcal{A}$ (in other words, the Lyapunov function does not depend on the uncertain parameter). The sufficient condition in Theorem 1 involves the matrices

$$\bar{S}_0(\beta) = \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_2 & \beta_3 \end{bmatrix}, \quad \bar{U}_2(\alpha) = \begin{bmatrix} 0 & 0 & 0 & -\alpha_1 & \alpha_3 & -\alpha_2-\alpha_4 \\ 0 & 0 & \alpha_1 & 0 & \alpha_2 & -\alpha_5 \\ 0 & \alpha_1 & 2\alpha_3 & \alpha_4 & 0 & -\alpha_6 \\ -\alpha_1 & 0 & \alpha_4 & 2\alpha_5 & \alpha_6 & 0 \\ -\alpha_3 & \alpha_2 & 0 & \alpha_6 & 0 & 0 \\ -\alpha_2-\alpha_4 & -\alpha_5 & -\alpha_6 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{R}_2(\beta) = \begin{bmatrix} r_1 & r_2 & 0 & 0 & 0 & 0 \\ r_2 & r_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_4 & r_5 & 0 & 0 \\ 0 & 0 & r_5 & r_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_7 & r_8 \\ 0 & 0 & 0 & 0 & r_8 & r_9 \end{bmatrix}$$

where

$$\begin{aligned} r_1 &= 2\beta_1 + (-8 + 26\rho - 28\rho^2)\beta_2 \\ r_2 &= (1 + 7\rho - 6\rho^2)\beta_1 + (2 - 3\rho + 2\rho^2)\beta_2 + (-4 + 13\rho - 14\rho^2)\beta_3 \\ r_3 &= (2 + 14\rho - 12\rho^2)\beta_2 + (2 - 6\rho + 4\rho^2)\beta_3 \\ r_4 &= 4\beta_1 + (-16 + 26\rho)\beta_2 \\ r_5 &= (2 + 7\rho)\beta_1 + (4 - 3\rho)\beta_2 + (-8 + 13\rho)\beta_3 \\ r_6 &= (4 + 14\rho)\beta_2 + (4 - 6\rho)\beta_3 \\ r_7 &= -2\beta_1 - 8\beta_2 \\ r_8 &= \beta_1 + 2\beta_2 - 4\beta_3 \\ r_9 &= 2\beta_2 + 2\beta_3. \end{aligned}$$

The number of free parameters is $u(2,2,2)+s(2,2,0) = 9$ and the lower bound of $\rho^*$ is $\rho^*_{\{m=0\}} = 0.22591$.

By using higher values of $m$ we find:

- $\rho^*_{\{m=1\}} = 0.76480$, with $u(2,2,3) + s(2,2,1) = 22$;
- $\rho^*_{\{m=2\}} = 1.3005$, with $u(2,2,4) + s(2,2,2) = 43$.

It turns out that the lower bound for $m = 2$ is tight, i.e. $\rho^*_{\{m=2\}} = \rho^*$. This is verified by the fact that $\hat{A}(\rho^*_{\{m=2\}})$ is marginally Hurwitz. The HPD-QLF matrix corresponding to $\rho^*_{\{m=2\}}$ is

$$P_2^*(p) = p_1^2 \begin{bmatrix} 1 & 0.0928 \\ 0.0928 & 0.0086 \end{bmatrix} + p_1 p_2 \begin{bmatrix} -0.7400 & 0.1072 \\ 0.1072 & 1.1183 \end{bmatrix} + p_2^2 \begin{bmatrix} 0.4849 & 0.0370 \\ 0.0370 & 0.2322 \end{bmatrix}.$$

### 5.2 Example 2

Consider the problem of computing the robust parametric margin $\rho^*$ defined as

$$\rho^* = \sup \left\{ \rho \in \mathbb{R} : \ \hat{A}(\theta_1, \theta_2) \text{ is Hurwitz for all } \theta = [\theta_1, \theta_2]' \in \rho\Theta \right\}$$

where the set $\Theta$ is chosen as in Figure 1 according to

$$\Theta = \left\{ \theta \in \mathbb{R}^2 : \ \theta_1 + \theta_2 \leq 1, \ \theta_i \geq 0 \right\}$$

and

$$\hat{A}(\theta_1, \theta_2) = \begin{bmatrix} -1 & -2 \\ 5 & 0 \end{bmatrix} + \theta_1 \begin{bmatrix} 0 & -5 \\ -15 & 1 \end{bmatrix} + \theta_2 \begin{bmatrix} -8 & -6 \\ -2 & 10 \end{bmatrix} + \theta_1^2 \begin{bmatrix} 0 & 6 \\ 14 & -2 \end{bmatrix} + \theta_2^2 \begin{bmatrix} 8 & 8 \\ 0 & -12 \end{bmatrix}.$$

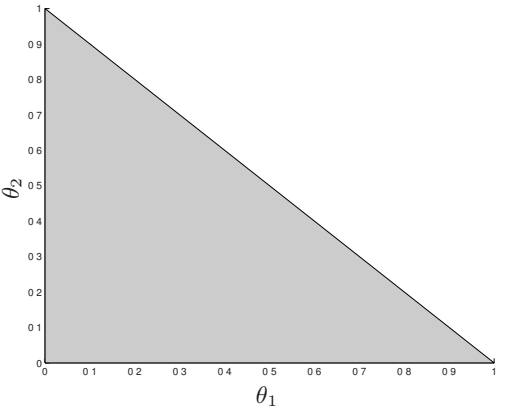



**Fig. 1.** Set $\Theta$.

The equivalent matrix $A(p;\rho)$ is computed by setting $\theta_1 = \rho p_1$ and $\theta_2 = \rho p_2$ and converting the so-obtained polynomial matrix in $p_1, p_2$ in a homogeneous one with respect to $p = [p_1, p_2, p_3]'$ where $p_3 = 1 - p_1 - p_2$. We hence have $q = 3$ and $n = r = 2$.

By using Theorem 1 we find:

- $\rho^*_{\{m=0\}} = 0.09825$ $(u(3,2,2) + s(3,2,0) = 36)$;

- $\rho^*_{\{m=1\}} = 0.38279$ $(u(3,2,3) + s(3,2,1) = 138)$;
- $\rho^*_{\{m=2\}} = 0.56349$ $(u(3,2,4) + s(3,2,2) = 381)$.

It turns out that the lower bound for $m = 2$ is tight, i.e. $\rho^*_{\{m=2\}} = \rho^*$. This has been verified in the following way. Define the degree of stability

$$\delta(\theta) = -\max\left\{\Re(\lambda) : \ \lambda \text{ is an eigenvalue of } \hat{A}(\theta)\right\}, \tag{31}$$

where $\Re(\lambda)$ denotes the real part of $\lambda$. Figure 2 shows the plot of $\delta(\theta)$ along the hypotenuse of $\rho^*_{\{m=2\}}\,\Theta$, i.e. the segment

$$\theta(\eta) = \begin{bmatrix} \rho^*_{\{m=2\}} \\ 0 \end{bmatrix} \eta + \begin{bmatrix} 0 \\ \rho^*_{\{m=2\}} \end{bmatrix} (1-\eta), \quad \eta \in [0,1].$$

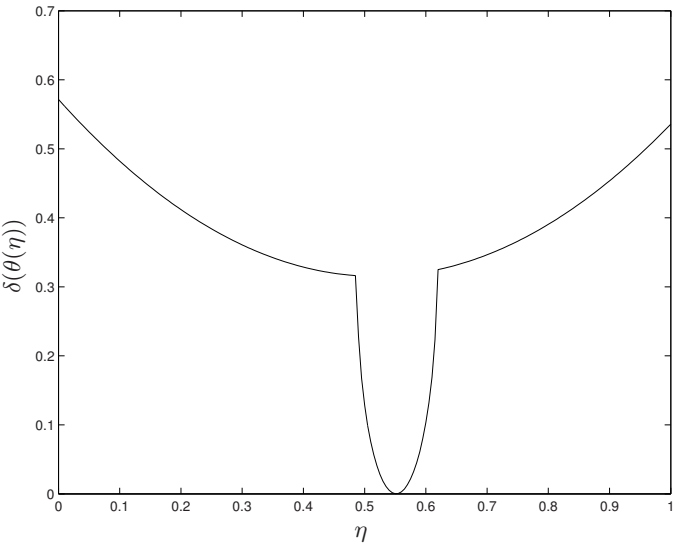



**Fig. 2.** Degree of stability $\delta(\theta)$ along the hypotenuse of the set $\rho^*_{\{m=2\}}\,\Theta$.

## 6 Conclusions

The class of HPD-QLFs has been shown to be a viable tool for assessing robust stability of uncertain linear systems. By expressing the dependence of the Lyapunov function on the uncertain parameters as a polynomial homogenous form, it is possible to formulate sufficient conditions in terms of LMI feasibility tests, which are less conservative with respect to conditions derived for linearly parameter-dependent Lyapunov functions.

Ongoing research concerns the application of HPD-QLFs to systems in which the uncertain parameters are allowed to be time-varying with a known bound on the variation rate, and to the evaluation of robust performance in control systems. Another topic of interest is the comparison of HPD-QLFs with a class of Lyapunov functions in which also the dependence on the state vector is expressed as a homogeneous polynomial form (not quadratic). Whether such class can provide less conservative conditions with respect to HPD-QLFs, for different uncertainty models, is still an open question.

## References

1. D. D. Šiljak (1969). Nonlinear Systems: Parametric Analysis and Design. John Wiley & Sons, New York.
2. D. D. Šiljak (1989). Parameter space methods for robust control design: a guided tour. IEEE Trans. on Automatic Control, 34:674–688.
3. B. R. Barmish (1994). New Tools for Robustness of Linear Systems. Mcmillan Publishing Company, New York.
4. S. P. Bhattacharyya, H. Chapellat, and L. H. Keel (1995). Robust Control: The Parametric Approach. Prentice Hall, NJ.
5. H. P. Horisberger and P. R. Belanger (1976). Regulators for linear time invariant plants with uncertain parameters. IEEE Trans. on Automatic Control, 21:705–708.
6. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia.
7. K. S. Narendra and J. H. Taylor (1973). Frequency domain criteria for absolute stability. Academic Press Inc., New York.
8. D. Peaucelle, D. Arzelier, O. Bachelier, and J. Bernussou (2000). A new robust $\mathcal{D}$-stability condition for real convex polytopic uncertainty. Systems and Control Letters, 40:21–30.
9. A. Trofino (1999). Parameter dependent Lyapunov functions for a class of uncertain linear systems: a LMI approach. Proc. IEEE Conf. on Decision and Control, 2341–2346, Phoenix, Arizona.
10. D. C. W. Ramos and P. L. D. Peres (2002). An LMI approach to compute robust stability domains for uncertain linear systems. IEEE Trans. on Automatic Control, 47:675–678.
11. V. J. S. Leite and P. L. D. Peres (2003). An improved LMI condition for robust $\mathcal{D}$-stability of uncertain polytopic systems. IEEE Trans. on Automatic Control, 48(3):500–504.
12. G. Chesi, A. Garulli, A. Tesi, and A. Vicino (2003). Robust stability for polytopic systems via polynomially parameter-dependent Lyapunov functions. Proc. IEEE Conf. on Decision and Control, Maui, Hawaii.
13. R. W. Brockett (1973). Lie algebra and Lie groups in control theory. In D.Q. Mayne and R.W. Brockett (Editors). Geometric Methods in Systems Theory, 43–82, Dordrecht, Reidel.
14. N. K. Bose (1982). Applied Multidimensional Systems Theory. Van Rostrand Reinhold, New York.

15. P. A. Parrilo (2000). Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology.
16. G. Chesi, A. Garulli, A. Tesi, and A. Vicino (2003). On the role of homogeneous forms in robustness analysis of control systems. In L. Giarré and B. Bamieh (Editors). Multidisciplinary Research in Control: The Mohammed Dahleh Symposium 2002, LNCIS 289:161–178, Springer-Verlag, Berlin.
17. G. Chesi, A. Garulli, A. Tesi, and A. Vicino (2003). Solving quadratic distance problems: an LMI-based approach. IEEE Trans. on Automatic Control, 48(2):200–212.
18. P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali (1995). LMI Control Toolbox for MATLAB. The Mathworks Inc.
19. J. F. Sturm (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 11–12:625–653.

# Stabilization of LPV Systems

Pierre-Alexandre Bliman

INRIA, Rocquencourt BP 105, 78153 Le Chesnay cedex, France
pierre-alexandre.bliman@inria.fr

We study here the static state-feedback stabilization of linear finite dimensional systems depending polynomially upon a finite set of real, bounded, parameters. These parameters are a priori unknown, but available in real-time for control. In consequence, it is natural to allow possible dependence of the gain with respect to the parameters (*gain-scheduling*).

We state two main results. First, we show that stabilizability of the class of systems obtained for frozen values of the parameters, may be expressed *equivalently* by linear matrix inequalities (LMIs), linked to certain class of parameter-dependent Lyapunov functions. Second, we show that existence of such a Lyapunov function for the linear parameter-varying (LPV) systems subject to bounded rate of variation of the parameters with respect to time, may be in the same manner expressed equivalently by LMI conditions. In both cases, the method provides explicitly parameter-dependent stabilizing gain. The central arguments are linked to the existence of a decomposition of some symmetric parameter-dependent matrices as sum of positive definite terms.

## 1 Introduction

Linear parameter-varying (LPV) systems have recently received much attention, in connection with the gain-scheduling control design methodologies, see [5, 12] for recent surveys and bibliography on the subject. LPV systems are linear systems that depend upon time-varying real parameters. The latter are not known in advance, but may be used in real-time for control purposes. However, they are usually constrained to lie inside a known bounded set.

The issue of checking the stabilizability and determining a parameter-dependent stabilizing gain for every frozen admissible value of the parameters, is already a difficult task. As an example, a coarse application of the Lyapunov-based synthesis techniques available for linear systems is impossible, as it leads to solve an infinite number of linear matrix inequalities (LMIs). At this point,

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 103–117, 2005.

two types of methods are usually used (see the recent works [14, 7] on LPV systems): either the controller gain is first computed for a bunch of parameter values, and then interpolated between the nodes of this grid (but the stability, and possibly performance, results are not guaranteed between the nodes); or the solution of the parameter-dependent LMIs involved is sought for with prespecified dependence with respect to the parameters, usually constant or affine (at the cost of adding conservatism). Of course, the stabilization issue is still more complicated when the parameters are time-varying.

In this paper, we show that, in principle, for linear systems depending polynomially upon finite number of bounded parameters, the determination of parameter-dependent stabilizing gain may be achieved without conservatism. More precisely, we state two main results (Theorems 1 and 2 below), whose contribution may be summarized as follows.

1. The stabilization of all the systems obtained for *constant values* of the parameters in the admissible hypercube is equivalent to the existence for the closed-loop system of a quadratic Lyapunov function *polynomial* with respect to the parameters. For fixed value of the degree, the coefficients of this polynomial may be found by solving a LMI.
2. The existence of a similar quadratic Lyapunov function (depending in the same way upon the parameters) for the corresponding LPV system with *restricted rate of variation* of the parameters, is also equivalent, for fixed degree, to the solvability of a LMI.
3. In both cases, a parameter-dependent stabilizing gain is deduced from the solution of the LMIs.

The originality of the results presented here lies in the nonconservative nature of the LMI conditions proposed. They constitute a systematization of the approaches based on parameter-dependent Lyapunov functions. Further work should consider dynamic controller synthesis and performace verification.

Effective use of the results given here is subordinate to powerful LMI solvers. A general idea for reducing the computation complexity consists in performing first a subdivision of the admissible parameter set in subdomains and applying the results presented below on these smaller regions. The present paper provides a stage towards such a *hybrid* control (with switches according to the parameter values), which in principle could lead to sensible diminution of the (off-line) computational burden, but whose study is out of our scope here.

The paper is organized as follows. The problem is presented in Sect. 2. Notations are provided in Sect. 3. The result on systems with frozen parameters (Theorem 1) is stated in Sect. 4. The results on systems with parameters with bounded derivative (Theorem 2) is stated in Sect. 5. Elements of proof are displayed in Sect. 6. Some technical results related to the computations involved are gathered in Appendix.

## 2 Problem Statement

We consider here the issue of state-feedback stabilization for the class of linear systems

$$\dot{x} = A(\sigma(t))x + B(\sigma(t))u \ . \tag{1}$$

In (1), the matrices $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times p}$ are supposed to be polynomials of partial degree (at most) $k$ with respect to the components of a vector $\sigma \stackrel{\text{def}}{=} (\sigma_1, \ldots, \sigma_m)$ of $m$ real parameters.

We are interested in the design of stabilizing static state-feedback for (1), under the assumption that $\forall t \geq 0$, $\sigma(t) \in [-1;+1]^m$. In the special case where the components of $\sigma$ are constant ($\dot{\sigma} \equiv 0$), this is equivalent to find, for any $\sigma \in [-1;+1]^m$, a gain $K(\sigma)$ such that $A(\sigma)+B(\sigma)K(\sigma)$ is Hurwitz. This leads to study the following property.

Property I. *There exist mappings* $P : [-1;+1]^m \to \mathcal{S}^n$, $N : [-1;+1]^m \to \mathbb{R}^{p\times n}$ *such that,* $\forall \sigma \in [-1;+1]^m$

$$P(\sigma) > 0_n, \ A(\sigma)P(\sigma) + P(\sigma)A(\sigma)^T + B(\sigma)N(\sigma) + N(\sigma)^T B(\sigma)^T < 0_n.$$

In this formula, $\mathcal{S}^n$ represents the set of symmetric matrices of size $n \times n$. Property I is *equivalent* to the stabilizability of (1) for every admissible choice of the parameters.

As is well-known, the previous condition, guaranteeing stability for the frozen parameter systems, is not enough to guarantee stability of the systems with time-varying parameters. An attempt to extend the previous ideas to stabilization of LPV systems with parameters having variation rate constrained by $|\dot{\sigma}_i| \leq \bar{\varrho}_i$ a.e., $i = 1, \ldots, m$, leads to the following interesting issue.

Property II. *There exist mappings* $P : [-1;+1]^m \to \mathcal{S}^n$, $N : [-1;+1]^m \to \mathbb{R}^{p\times n}$, *$P$ differentiable, such that,* $\forall \sigma \in [-1;+1]^m$, $\forall \varrho_i \in [-\bar{\varrho}_i; \bar{\varrho}_i]$,

$$P(\sigma) > 0_n, \ A(\sigma)P(\sigma)+P(\sigma)A(\sigma)^T+B(\sigma)N(\sigma)+N(\sigma)^T B(\sigma)^T - \sum_{i=1}^{m} \varrho_i \frac{\partial P(\sigma)}{\partial \sigma_i} < 0_n.$$

Property II is *equivalent* to the existence of a quadratic Lyapunov function depending regularly upon the present values of the parameters. This property is thus a priori stronger than the stability of the systems (1) attached to every admissible trajectories of the parameters.

## 3 Notations and Preliminaries

• The matrices $I_n$, $0_n$, $0_{n\times p}$ are the $n \times n$ identity matrix and the $n \times n$ and $n \times p$ zero matrices respectively. The symbol $\otimes$ denotes Kronecker product, the power of Kronecker product being used with the natural meaning: $M^{0\otimes} = 1$, $M^{p\otimes} \stackrel{\text{def}}{=} M^{(p-1)\otimes} \otimes M$. Key properties are: $(A \otimes B)^T = A^T \otimes B^T$, $(A \otimes$

$B)(C \otimes D) = (AC \otimes BD)$ for matrices of compatible size. The transpose and transconjugate of $M$ are denoted $M^T$ and $M^*$. The unit circle in $\mathbb{C}$ is denoted as the boundary $\partial\mathbb{D}$, and the set of positive integers $\mathbb{N}$. Diagonal matrices are defined by diag. Also, the set of symmetric real (resp. hermitian complex) matrices of size $n \times n$ is denoted $\mathcal{S}^n$ (resp. $\mathcal{H}^n$).

Last, we introduce some spaces of matrix-valued polynomials. $\mathbb{R}^{n\times n}[\sigma]$ (resp. $\mathcal{S}^n[\sigma]$) will denote the set of polynomials in the variable $\sigma \in \mathbb{R}^m$, with coefficients in $\mathbb{R}^{n\times n}$ (resp. $\mathcal{S}^n$). We shall also consider in the sequel the set, denoted $\mathbb{R}^{n\times n}[z,\bar{z}]$, of polynomials in $z$ and $\bar{z}$, $z \in \mathbb{C}$, with coefficients in $\mathbb{R}^{n\times n}$. The sets $\mathcal{S}^n[z,\bar{z}]$, $\mathcal{H}^n[z,\bar{z}]$ are defined similarly.

• We now introduce specific notations. For any $l \in \mathbb{N}$, for any $v \in \mathbb{C}$, let

$$v^{[l]} \stackrel{\text{def}}{=} \begin{pmatrix} 1 \\ v \\ \vdots \\ v^{l-1} \end{pmatrix} . \tag{2}$$

This notation permits to manipulate polynomials. Notice in particular that, for a free variable $z \in \mathbb{C}^m$, the vector $(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]})$ contains exactly the $l^m$ monomials in $z_1, \ldots, z_m$ of degree at most $l-1$ in each variable.

Using this notation, any element $M(z)$ in $\mathbb{R}^{p\times n}[z,\bar{z}]$ may be represented as

$$M(z) = (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_p)^* M_l (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) . \tag{3}$$

In this formula, for given $l \in \mathbb{N}$, the matrix $M_l \in \mathbb{R}^{l^m p \times l^m n}$ is unique, in the sense that: $M(z) = 0$ *for all* $z \in \mathbb{C}^m$ *iff* $M_l = 0$. Independently of minimality, the matrix $M_l$ is called the *coefficient matrix* of this representation of $M(z)$, $l-1$ its *degree*.

In the sequel, we shall use the following *change of variables* ($\mathrm{i}^2 = -1$):

$$\begin{gathered} \varphi : [-1;+1]^m \to (\partial\mathbb{D})^m, \sigma \mapsto z = \varphi(\sigma) \\ \text{where } z_i \stackrel{\text{def}}{=} \sigma_i + \mathrm{i}\sqrt{1-\sigma_i^2}, i = 1, \ldots, m . \end{gathered} \tag{4}$$

Basically (see the developments below), changing $\sigma$ in $z$ will permit to use Kalman-Yakubovich-Popov lemma, "replacing" the free variables $z_i$ by matrix multipliers in the parameter-dependent LMIs appearing in Properties I and II. In particular, for $z$ in the range of $\varphi$, $\varphi^{-1}(z) = \frac{z+\bar{z}}{2}$. When $z = (z_1, \ldots, z_m)$ covers $(\partial\mathbb{D})^m$, $\frac{z+\bar{z}}{2}$ varies in the whole set $[-1;+1]^m$.

Generally speaking, for $M$ defined as in (3) and the change of variable $\varphi$ as in (4), $M(\varphi(\sigma))$ is a polynomial in $\sigma_i$ and $\sqrt{1-\sigma_i^2}$, $i = 1, \ldots, m$. Among these polynomials, some will be of particular interest here, those leading to polynomials in the $\sigma_i$ *only*. It may checked easily that these are the polynomials whose coefficients in the monomials $\prod_{i=1,\ldots,m} z_i^{\alpha_i} \bar{z}_i^{\alpha_i'}$ and $\prod_{i=1,\ldots,m} z_i^{\beta_i} \bar{z}_i^{\beta_i'}$

are equal when $\{\alpha_i, \alpha'_i\} = \{\beta_i, \beta'_i\}$ for any $i = 1, \ldots, m$. Indeed, up to factorization by powers of $|z_i|^2$ (which is equal to 1 on $\partial\mathbb{D}$), those polynomials are functions of $z_i + \bar{z}_i = 2\sigma_i$ only. This property corresponds to matrices $M_l \in \mathbb{R}^{l^m p \times l^m n}$ in (3) having a particular *mirror* block structure, those pertaining to the set

$$\mathbb{R}_M^{p\times n, l^m} \stackrel{\text{def}}{=} \{M_l \in \mathbb{R}^{l^m p \times l^m n} \ : \ \forall \alpha_1, \ldots, \alpha_m, \alpha'_1, \ldots, \alpha'_m \in \{0, \ldots, l-1\},$$
$$(e_{\alpha_m} \otimes \cdots \otimes e_{\alpha_1} \otimes I_p)^T M_l (e_{\alpha'_m} \otimes \cdots \otimes e_{\alpha'_1} \otimes I_n) \text{ depends only upon the sets}$$
$$\{\alpha_1, \alpha'_1\}, \ldots, \{\alpha_m, \alpha'_m\}\} \ ,$$

where we put $e_\alpha^T \stackrel{\text{def}}{=} \left(0_{1\times\alpha}\ 1\ 0_{1\times(l-\alpha-1)}\right)$.

The definition of $\mathbb{R}_M^{p\times n, l^m}$ is such that $M_l \in \mathbb{R}_M^{p\times n, l^m}$ *iff* $M(\varphi(\sigma))$ is polynomial in $\sigma \in [-1; +1]^m$, for $M(z)$ defined by (3). The subset of those maps $M(z)$ of $\mathbb{R}^{p\times n}[z, \bar{z}]$ such that $M(\varphi(\sigma))$ is polynomial in $\sigma \in [-1; +1]^m$, will be denoted $\mathbb{R}_M^{p\times n}[z, \bar{z}]$. Also, we define $\mathcal{S}_M^n[z, \bar{z}] \stackrel{\text{def}}{=} \mathbb{R}_M^{n\times n}[z, \bar{z}] \cap \mathcal{S}^n[z, \bar{z}]$.

Let us point out to the reader, that some technical results linked to the matrix transformations induced by operations on polynomials, are gathered in Appendix.

• We finally define some matrices. For $l, l' \in \mathbb{N}$, let $\hat{J}_{l',l}, \check{J}_{l',l} \in \mathbb{R}^{l\times(l+l')}$ be defined by

$$\hat{J}_{l',l} \stackrel{\text{def}}{=} \left(I_l\ 0_{l\times l'}\right), \ \check{J}_{l',l} \stackrel{\text{def}}{=} \left(0_{l\times l'}\ I_l\right) \ . \tag{5}$$

A key property of these matrices is that, $\forall v \in \mathbb{C}$, for $v^{[l]}$ defined previously,

$$v^{[l]} = \hat{J}_{l',l} v^{[l+l']}, \ v^{l'} v^{[l]} = \check{J}_{l',l} v^{[l+l']} \ . \tag{6}$$

Last, define $L_l \in \mathbb{R}^{l\times l}$ by:

$$L_l \stackrel{\text{def}}{=} \left(\begin{array}{ccc|c} 0 & \ldots & 0 & 0 \\ \hline 1 & & & 0 \\ & 2 & & \\ & & \ddots & \vdots \\ & & l-1 & 0 \end{array}\right) \ . \tag{7}$$

## 4 Constant Parameters

In the case where the parameters $\sigma$ are *constant*, it turns out that Property I is fulfilled *if and only if* it is fulfilled for certain $P(\sigma), N(\sigma)$ depending polynomially upon $\sigma$ (see also [2]). This naturally introduces as new variables the degree $l-1$ of the polynomials, and the coefficient matrices of $P$ and $N$.

It turns out moreover, that, for given $l$, the coefficients may be found out by solving an LMI. This permits to find in an explicit way stabilizing controllers, as functions of the parameter $\sigma$.

**Theorem 1.** *The following assertions are* equivalent*.*

*(i) Property* I *is fulfilled.*
*(ii) There exists* $(P(\sigma), N(\sigma)) \in \mathcal{S}^n[\sigma] \times \mathbb{R}^{p\times n}[\sigma]$ *fulfilling Property* I.
*(iii) There exist an integer* $l \in \mathbb{N}$*,* 2 *matrices* $P_l \in \mathcal{S}^{l^m n} \cap \mathbb{R}_M^{n\times n, l^m}$, $N_l \in \mathbb{R}_M^{p\times n, l^m}$ *and* $2m$ *matrices* $Q_{l,i}^P \in \mathcal{S}^{(l-1)^{m-i+1} l^{i-1} n}$, $Q_{l,i}^R \in \mathcal{S}^{(k+l-1)^{m-i+1}(k+l)^{i-1} n}$, $i = 1, \ldots, m$*, such that the system* (8) *of* 2 *LMIs is fulfilled, where* $R_{k+l} = R_{k+l}(P_l, N_l) \in \mathcal{S}^{(k+l)^m n}$ *is the coefficient matrix of* $R(z)$ *defined in* (9)*, corresponding to* $P(z), N(z)$ *with coefficient matrices* $P_l, N_l$*.*

$$0_{l^m n} < P_l + \sum_{i=1}^{m} \left(\hat{J}_{1,l-1}^{(m-i+1)\otimes} \otimes I_{l^{i-1}n}\right)^T Q_{l,i}^P \left(\hat{J}_{1,l-1}^{(m-i+1)\otimes} \otimes I_{l^{i-1}n}\right)$$
$$- \sum_{i=1}^{m} \left(\hat{J}_{1,l-1}^{(m-i)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{i-1}n}\right)^T Q_{l,i}^P \left(\hat{J}_{1,l-1}^{(m-i)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{i-1}n}\right) , \tag{8a}$$

$$0_{(k+l)^m n} > R_{k+l} + \sum_{i=1}^{m} \left(\hat{J}_{1,k+l-1}^{(m-i+1)\otimes} \otimes I_{(k+l)^{i-1}n}\right)^T Q_{l,i}^R \left(\hat{J}_{1,k+l-1}^{(m-i+1)\otimes} \otimes I_{(k+l)^{i-1}n}\right)$$
$$- \sum_{i=1}^{m} \left(\hat{J}_{1,k+l-1}^{(m-i)\otimes} \otimes \check{J}_{1,k+l-1} \otimes I_{(k+l)^{i-1}n}\right)^T Q_{l,i}^R \left(\hat{J}_{1,k+l-1}^{(m-i)\otimes} \otimes \check{J}_{1,k+l-1} \otimes I_{(k+l)^{i-1}n}\right), \tag{8b}$$

$$R(z) \stackrel{\text{def}}{=} A(\frac{z+\bar{z}}{2})P(z) + P(z)A(\frac{z+\bar{z}}{2})^T + B(\frac{z+\bar{z}}{2})N(z) + N(z)^T B(\frac{z+\bar{z}}{2})^T < 0_n . \tag{9}$$

*Moreover,*

- *given a solution of LMI* (8)*, for* $P(z), N(z)$ *having coefficient matrices* $P_l, N_l$*,* $P(\varphi(\sigma)), N(\varphi(\sigma))$ *fulfil Property* I*, and* $K(\sigma) \stackrel{\text{def}}{=} N(\varphi(\sigma))P(\varphi(\sigma))^{-1}$ *is a stabilizing gain, rational in* $\sigma$*;*
- *if LMI* (8) *is solvable for the value* $l$ *of the index, then it is also solvable for any larger value.*

The matrices $\hat{J}, \check{J}$ have been defined earlier in (5). Details for a systematic computation of the matrix $R_{k+l}$ and of the gain $K(\sigma)$ may be found in Appendix.

Theorem 1 offers a family of relaxations of Property I. These conditions are less and less conservative when the index $l$ increases. Asymptotically, the

conservatism vanishes, as solvability of (8) for certain $l$ is also *necessary* to have Property I.

Notice that the two inequalities in (8) correspond respectively to the conditions

$$P(\frac{z+\bar{z}}{2}) > 0_n$$

and

$$R(\frac{z+\bar{z}}{2}) = A(\frac{z+\bar{z}}{2})P(z)+P(z)A(\frac{z+\bar{z}}{2})^T+B(\frac{z+\bar{z}}{2})N(z)+N(z)^TB(\frac{z+\bar{z}}{2})^T < 0_n$$

for all $z \in (\partial\mathbb{D})^m$. Elements of proof of Theorem 1 are provided in Sect. 6, but we briefly indicate here how to prove that feasibility of (8) implies Property I. Right- and left-multiplication of (8a) by $(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)$ and its transconjugate yields, using (6) repeatedly:

$$\begin{aligned} 0_n < (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)^* P_l (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) + \sum_{i=1}^{m} (1-|z_i|^2) \\ \times (z_m^{[l-1]} \otimes \cdots \otimes z_i^{[l-1]} \otimes z_{i-1}^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)^* Q_{l,i}^P \\ \times (z_m^{[l-1]} \otimes \cdots \otimes z_i^{[l-1]} \otimes z_{i-1}^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) \ , \end{aligned}$$

from which one deduces, putting $|z_i| = 1$:

$$\forall z \in (\partial\mathbb{D})^m,\ P(z) \stackrel{\text{def}}{=} (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)^* P_l (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) > 0_n \ .$$

Applying similar argument on (8b) with $(z_m^{[k+l]} \otimes \cdots \otimes z_1^{[k+l]} \otimes I_n))$ leads to

$$\forall z \in (\partial\mathbb{D})^m,\ R(z) \stackrel{\text{def}}{=} (z_m^{[k+l]} \otimes \cdots \otimes z_1^{[k+l]} \otimes I_n)^* R_{k+l} (z_m^{[k+l]} \otimes \cdots \otimes z_1^{[k+l]} \otimes I_n) < 0_n \ .$$

It is now evident that solvability of (8) gives rise to a solution $(P, N)$ of Problem I of degree $l-1$ in $z, \bar{z}$, and $K(\sigma)$ as defined in the statement appears as a stabilizing gain, for every admissible value of the parameters.

Remark that, writing the positive right-hand side of, say, (8a) as $U^T \Lambda U$ with $U^T = U^{-1}$ and $\Lambda = \text{diag}\{\Lambda_i\}$, the previous computations show that, for any $z \in (\partial\mathbb{D})^m$,

$$P(z) = \sum_{i=1}^{l^m n} \Lambda_i \left( U(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) \right)_i^* \left( U(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) \right)_i \ ,$$

which thus appears as a sum of squares of matrix-valued polynomials.

Incidentally, stabilizability of a pair $(A, B)$ is equivalent [4, §7.2.1] to the existence of a definite positive matrix $P$ such that $AP + PA^T < BB^T$. This corresponds to the choice $N = -\frac{1}{2}B^T$ in the LMI: $AP+PA^T+BN+N^TB^T < 0$. Similarly, it may be checked that, replacing in (9) the matrix $N(z)$ by $-\frac{1}{2}B^T(\frac{z+\bar{z}}{2})$, provides a simpler stabilizability criterion. Another particular case is $B(\sigma) = 0$, which provides a *robust stability* criterion, see also [1].

## 5 Time-Varying Parameters with Bounded Variation

Contrary to the constant-parameter case, when Property II is fulfilled, there is probably no necessity for existence of a parameter-dependent Lyapunov function of the kind exhibited in Theorem 1. See however related results in [8, 9, 10]. But it is worth noting that, for given degree, the existence of such a Lyapunov function may be expressed without loss of generality as a LMI problem, in a way similar to what was done for Property I in Theorem 1. Analogously, stabilizing controllers are then found explicitly as functions of $\sigma(t)$.

**Theorem 2.** *The following assertions are* equivalent*.*

*(i) There exists* $(P(\sigma), N(\sigma)) \in \mathcal{S}^n[\sigma] \times \mathbb{R}^{p\times n}[\sigma]$ *fulfilling Property* II*.*

*(ii) There exist an integer* $l \in \mathbb{N}$*, 2 matrices* $P_l \in \mathcal{S}^{l^m n} \cap \mathbb{R}_M^{n\times n, l^m}$*,* $N_l \in \mathbb{R}_M^{p\times n, l^m}$*, m matrices* $Q_{l,i}^P \in \mathcal{S}^{(l-1)^{m-i+1} l^{i-1} n}$*,* $i = 1, \ldots, m$*, and* $2^m$ *matrices* $Q_{l,i}^{R,\eta} \in \mathcal{S}^{(k+l-1)^{m-i+1}(k+l)^{i-1} n}$*,* $i = 1, \ldots, m$*,* $\eta \in \{-1, 1\}^m$ *such that the system* (10) *of* $(2^m+1)$ *LMIs obtained for all* $\eta$ *in* $\{-1, 1\}^m$ *is fulfilled, where* $R_{k+l} = R_{k+l}(P_l, N_l)$ *has the same meaning than in Theorem* 1 *and* $\hat{P}_{k+l,i} \in \mathcal{S}^{(k+l)^m n}$ *is a coefficient matrix of the map* $z \mapsto \frac{\partial P(\varphi(\sigma))}{\partial \sigma_i}|_{\sigma=\frac{z+\bar{z}}{2}}$*.*

$$0_{l^m n} < P_l + \sum_{i=1}^{m} \left(\hat{J}_{1,l-1}^{(m-i+1)\otimes} \otimes I_{l^{i-1}n}\right)^T Q_{l,i}^P \left(\hat{J}_{1,l-1}^{(m-i+1)\otimes} \otimes I_{l^{i-1}n}\right)$$
$$- \sum_{i=1}^{m} \left(\hat{J}_{1,l-1}^{(m-i)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{i-1}n}\right)^T Q_{l,i}^P \left(\hat{J}_{1,l-1}^{(m-i)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{i-1}n}\right), \quad (10a)$$

$$0_{(k+l)^m n} > R_{k+l} + \sum_{i=1}^{m} \eta_i \bar{\varrho}_i \hat{P}_{k+l,i}$$
$$+ \sum_{i=1}^{m} \left(\hat{J}_{1,k+l-1}^{(m-i+1)\otimes} \otimes I_{(k+l)^{i-1}n}\right)^T Q_{l,i}^{R,\eta} \left(\hat{J}_{1,k+l-1}^{(m-i+1)\otimes} \otimes I_{(k+l)^{i-1}n}\right)$$
$$- \sum_{i=1}^{m} \left(\hat{J}_{1,k+l-1}^{(m-i)\otimes} \otimes \check{J}_{1,k+l-1} \otimes I_{(k+l)^{i-1}n}\right)^T Q_{l,i}^{R,\eta} \left(\hat{J}_{1,k+l-1}^{(m-i)\otimes} \otimes \check{J}_{1,k+l-1} \otimes I_{(k+l)^{i-1}n}\right), \quad (10b)$$

*Moreover,*

- *given a solution of LMI* (10)*, for* $P(z), N(z)$ *having coefficient matrices* $P_l, N_l$*,* $P(\varphi(\sigma)), N(\varphi(\sigma))$ *fulfil Property* II*, and, for any absolutely continuous* $\sigma$ *such that* $\sigma(t) \in [-1; +1]^m$*,* $\dot{\sigma}(t) \in \prod_{i=1}^m [-\bar{\varrho}_i; +\bar{\varrho}_i]$ *almost everywhere,*

$$K(\sigma(t)) \stackrel{\text{def}}{=} N(\varphi(\sigma(t)))P(\varphi(\sigma(t)))^{-1}$$

*is a stabilizing gain, rational in* $\sigma(t)$*;*

- *if LMI* (10) *is solvable for the value* $l$ *of the index, then it is also solvable for any larger value.*

The LMIs in Theorems 1 and 2 differ only by the presence of the terms in $\hat{P}_{k+l,i}$ in (10b). The latter correspond to the derivative terms $\frac{\partial P(\sigma)}{\partial \sigma_i}$ appearing in the inequality in Property II. See Appendix for details on the computations.

## 6 Elements of Demonstration

We only give here indications for proving Theorems 1 and 2. Application of the same techniques may be found in [3, 1], under more detailed form.

### 6.1 Sketch of Proof of Theorem 1

**1.** The equivalence between (i) and (ii), i.e. the fact that $P$, $N$ in Property I may be supposed polynomial without loss of generality, is consequence of a result on existence of polynomial solutions for LMIs depending continuously upon parameters lying in a compact set, see [2].
**2.** Take now (ii) as departure: there exists $(P, N) \in \mathcal{S}_M^n[z, \bar{z}] \times \mathbb{R}_M^{p\times n}[z, \bar{z}]$, with coefficient matrices $P_l \in \mathcal{S}^{l^m n} \cap \mathbb{R}_M^{n\times n, l^m}$, $N_l \in \mathbb{R}_M^{p\times n, l^m}$ for a certain integer $l$, such that, $\forall z \in (\partial\mathbb{D})^m$, $(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)^* P_l (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n) > 0_n$ and $(z_m^{[k+l]} \otimes \cdots \otimes z_1^{[k+l]} \otimes I_n)^* R_{k+l}(z_m^{[k+l]} \otimes \cdots \otimes z_1^{[k+l]} \otimes I_n) < 0_n$, for $R_{k+l}(P_l, N_l)$ defined as in the statement. The proof consists in achieving joint reduction of these two inequalities to the LMIs in (8). For simplicity, we expose this procedure for one inequality only, the first one. For $i = 0, \ldots, m$, denote $(\mathcal{P}_i)$ the property: $\exists l \in \mathbb{N}, \exists Q_{l,1}^P \in \mathcal{H}^{(l-1)^m n}$, $\ldots$, $\exists Q_{l,i}^P \in \mathcal{H}^{(l-1)^{m-i+1} l^{i-1} n}$, $\forall (z_{i+1}, \ldots, z_m) \in (\partial\mathbb{D})^{m-i}$ such that (11) holds:

$$\left(z_m^{[l]} \otimes \cdots \otimes z_{i+1}^{[l]} \otimes I_{l^i n}\right)^* \left[ P_l + \sum_{j=1}^{i} \left(\hat{J}_{1,l-1}^{(m-j+1)\otimes} \otimes I_{l^{j-1} n}\right)^T Q_{l,j}^P \left(\hat{J}_{1,l-1}^{(m-j+1)\otimes} \otimes I_{l^{j-1} n}\right) \right.$$
$$\left. - \sum_{j=1}^{i} \left(\hat{J}_{1,l-1}^{(m-j)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{j-1} n}\right)^T Q_{l,j}^P \left(\hat{J}_{1,l-1}^{(m-j)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{j-1} n}\right) \right]$$
$$\times \left(z_m^{[l]} \otimes \cdots \otimes z_{i+1}^{[l]} \otimes I_{l^i n}\right) > 0_{l^i n} \,. \tag{11}$$

Property $(\mathcal{P}_0)$ is the part of (ii) devoted to $P$, whereas $(\mathcal{P}_m)$ is just (8a). We indicate in the remaining, how to establish that $(\mathcal{P}_i) \Leftrightarrow (\mathcal{P}_{i+1})$ for any $i = 0, \ldots, m-1$.

Remark that

$$(z_m^{[l]} \otimes \cdots \otimes z_{i+1}^{[l]} \otimes I_{l^i n}) = (z_m^{[l]} \otimes \cdots \otimes z_{i+2}^{[l]} \otimes I_{l^{i+1} n})(z_{i+1}^{[l]} \otimes I_{l^i n})$$

and

$$(z_{i+1}^{[l]} \otimes I_{l^i n}) = \begin{pmatrix} I_{l^i n} \\ z_{i+1} \left( I_{(l-1)l^i n} - z_{i+1}(F_{l-1} \otimes I_{l^i n}) \right)^{-1} (f_{l-1} \otimes I_{l^i n}) \end{pmatrix}$$

with

$$F_l \stackrel{\text{def}}{=} \begin{pmatrix} 0_{1\times(l-1)} & 0 \\ I_{l-1} & 0_{(l-1)\times 1} \end{pmatrix}, \quad f_l \stackrel{\text{def}}{=} \begin{pmatrix} 1 \\ 0_{(l-1)\times 1} \end{pmatrix}.$$

Applying discrete-time Kalman-Yakubovich-Popov lemma (see [13, 11] and the statement in the complex case for the continuous-time case in [6, Theorem 1.11.1 and Remark 1.11.1]) yields equivalence of $(\mathcal{P}_i)$ with: $\exists l \in \mathbb{N}, \exists Q_{l,1}^P \in \mathcal{H}^{(l-1)^m n}, \ldots, \exists Q_{l,i}^P \in \mathcal{H}^{(l-1)^{m-i+1} l^{i-1} n}$, $\forall (z_{i+2}, \ldots, z_m) \in (\partial\mathbb{D})^{m-i-1}$, $\exists \tilde{Q}_{l,i+1}^P(z_{i+2}, \ldots, z_m) \in \mathcal{H}^{(l-1)l^i n}$ such that:

$$\begin{aligned} 0_{l^{i+1}n} < \left(z_m^{[l]} \otimes \cdots \otimes z_{i+2}^{[l]} \otimes I_{l^{i+1}n}\right)^* \Bigg[ & P_l \\ & + \sum_{j=1}^{i} \left(\hat{J}_{1,l-1}^{(m-j+1)\otimes} \otimes I_{l^{j-1}n}\right)^T Q_{l,j}^P \left(\hat{J}_{1,l-1}^{(m-j+1)\otimes} \otimes I_{l^{j-1}n}\right) \\ & - \sum_{j=1}^{i} \left(\hat{J}_{1,l-1}^{(m-j)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{j-1}n}\right)^T Q_{l,j}^P \left(\hat{J}_{1,l-1}^{(m-j)\otimes} \otimes \check{J}_{1,l-1} \otimes I_{l^{j-1}n}\right) \Bigg] \\ & \times \left(z_m^{[l]} \otimes \cdots \otimes z_{i+2}^{[l]} \otimes I_{l^{i+1}n}\right) \\ + \left(\hat{J}_{1,l-1} \otimes I_{l^i n}\right)^T \tilde{Q}_{l,i+1}^P & \left(\hat{J}_{1,l-1} \otimes I_{l^i n}\right) - \left(\check{J}_{1,l-1} \otimes I_{l^i n}\right)^T \tilde{Q}_{l,i+1}^P \left(\check{J}_{1,l-1} \otimes I_{l^i n}\right). \end{aligned} \tag{12}$$

**3.** Using again the result in [2], $\tilde{Q}_{l,i+1}^P(z_{i+2}, \ldots, z_m)$, solution of a LMI with parameter in $(\partial\mathbb{D})^{m-i-1}$, may be chosen polynomial in its variables and their conjugates. Let $\tilde{l} - 2$ be its degree. If $\tilde{l} \leq l$, then $\tilde{Q}_{l,i+1}^P(z_{i+2}, \ldots, z_m) = (z_m^{[l-1]} \otimes \cdots \otimes z_{i+2}^{[l-1]} \otimes I_{(l-1)l^i n})^* Q_{l,i+1}^P (z_m^{[l-1]} \otimes \cdots \otimes z_{i+2}^{[l-1]} \otimes I_{(l-1)l^i n})$, for a coefficient matrix $Q_{l,i+1}^P \in \mathcal{H}^{(l-1)^{m-i} l^i n}$. If $\tilde{l} > l$, it may be shown that, *up to an increase of* $l$, the degree may be supposed the same, so same formula holds (see [3, 1] for similar arguments).

At this point, the last two terms in inequality (12) have been transformed in:

$$\left(\hat{J}_{1,l-1}\otimes I_{l^in}\right)^T\left(z_m^{[l-1]}\otimes\cdots\otimes z_{i+2}^{[l-1]}\otimes I_{(l-1)l^in}\right)^* Q^P_{l,i+1}$$
$$\times\left(z_m^{[l-1]}\otimes\cdots\otimes z_{i+2}^{[l-1]}\otimes I_{(l-1)l^in}\right)\left(\hat{J}_{1,l-1}\otimes I_{l^in}\right)$$
$$-\left(\check{J}_{1,l-1}\otimes I_{l^in}\right)^T\left(z_m^{[l-1]}\otimes\cdots\otimes z_{i+2}^{[l-1]}\otimes I_{(l-1)l^in}\right)^* Q^P_{l,i+1}$$
$$\times\left(z_m^{[l-1]}\otimes\cdots\otimes z_{i+2}^{[l-1]}\otimes I_{(l-1)l^in}\right)\left(\check{J}_{1,l-1}\otimes I_{l^in}\right)\ ,$$

for a certain matrix $Q^P_{l,i+1}\in\mathcal{H}^{(l-1)^{m-i}l^in}$.

**4.** Some matrix intervertions in the last two terms of the previous formula finally yields equivalence between $(\mathcal{P}_i)$ and $(\mathcal{P}_{i+1})$.

**5.** The assertion that solvability of (8) for index $l$ implies the same property for every larger index, is proved using the same techniques than the one evoked (but not displayed) in point **3.**, to increase the size of the solution.

**6.** Last, the same argument is applied to (8b), with detail variations. Application to (8a) and (8b) has to be done together, because of the coupling term $P_l$. Due to the fact that solvability of (8a), resp. (8b), for a value $l$ of the index implies solvability for every larger value, taking a value for which both inequalities are solvable yields equivalence of (ii) and (iii).

### 6.2 Sketch of Proof of Theorem 2

The demonstration is copied from the demonstration of the previous Theorem. Due to the affine dependence upon the $\varrho_i$ in Property II, it is enough to consider only the extremal values $\pm\bar{\varrho}_i$. It is hence required that: $\exists l\in\mathbb{N}$, $\exists P_l\in\mathcal{S}^{l^mn}$, $\exists N_l\in\mathbb{R}_M^{p\times n,l^m}$, $\forall\eta\in\{-1,1\}^m$, $\forall z\in(\partial\mathbb{D})^m$, $(z_m^{[l]}\otimes\cdots\otimes z_1^{[l]}\otimes I_n)^*P_l(z_m^{[l]}\otimes\cdots\otimes z_1^{[l]}\otimes I_n)<0_n$ and $(z_m^{[k+l]}\otimes\cdots\otimes z_1^{[k+l]}\otimes I_n)^*\left(R_{k+l}+\sum_{i=1}^m\eta_i\bar{\varrho}_i\hat{P}_{k+l,i}\right)(z_m^{[k+l]}\otimes\cdots\otimes z_1^{[k+l]}\otimes I_n)<0_n$.

The argument then essentially follows the proof of Theorem 1. One has to check carefully that the process of increase of the degree (point **3.** in Sect. 6.1) still works.

## A Appendix on Polynomial Matrices

We give here details on the computations necessary for systematic use of Theorems 1 and 2. It is explained in Sects. A.1 and A.2 how to compute $R_{k+l}(P_l,N_l)$, that is how to determine the coefficient matrices of the terms in (9). Then in Sect. A.3 are provided formulas for explicit computation of $K(\sigma)$ as a function of $\sigma$, that is of $P(\varphi(\sigma))$ and $N(\varphi(\sigma))$ for $P(z),N(z)$ defined by their coefficient matrix $P_l,N_l$. Last, the computation of the term $\hat{P}_{k+l,i}$ in (10) is explained in Sect. A.4.

We first extend the notations defined in (5). For $l,l'\in\mathbb{N}$, $l\le l'$, $\alpha=0,1,\ldots,l'$, define $J_{\alpha,l,l'}\in\mathbb{R}^{l\times(l+l')}$ by:

$$J_{\alpha,l,l'} \stackrel{\text{def}}{=} \left(0_{l\times\alpha}\ I_l\ 0_{l\times(l'-\alpha)}\right)\ .$$

Then $\hat{J}_{l',l} = J_{0,l,l'}$, $\check{J}_{l',l} = J_{l',l,l'}$, and $v^\alpha v^{[l]} = J_{\alpha,l,l'} v^{[l+l']}$.

### A.1 Representation of Polynomial Matrices

A rather natural representation for a matrix-valued polynomial $M : \mathbb{R}^m \to \mathbb{R}^{p\times n}$ (such as $A(\sigma)$ and $B(\sigma)$) of degree $l-1$ is

$$M(\sigma) = \tilde{M}_l(\sigma_m^{[l]} \otimes \cdots \otimes \sigma_1^{[l]} \otimes I_n)\ , \tag{13}$$

for a certain, uniquely defined, matrix $\tilde{M}_l \in \mathbb{R}^{p\times l^m n}$. From this, one should be able to deduce the coefficient matrix of the map $M(\frac{z+\bar{z}}{2})$, in order to apply Theorems 1 and 2. The effect of the corresponding change of variable (4) is summarized by Lemma 1.

**Lemma 1.** *Let $\tilde{M}_l \in \mathbb{R}^{p\times l^m n}$, then $\tilde{M}_l \left(\left(\frac{z_m+\bar{z}_m}{2}\right)^{[l]} \otimes \cdots \otimes \left(\frac{z_1+\bar{z}_1}{2}\right)^{[l]} \otimes I_n\right) = (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_p)^* M_l (z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)$, where the matrix $M_l \in \mathbb{R}_M^{p\times n, l^m}$ is given by the formula $M_l \stackrel{\text{def}}{=} \sum_{0\le\alpha_i\le l-1} (J_{\alpha_m,1,l-1} \otimes \cdots \otimes J_{\alpha_1,1,l-1} \otimes I_p)^T \tilde{M}_l (K_{l,\alpha_m} \otimes \cdots \otimes K_{l,\alpha_1} \otimes I_n)$, in which, by definition, the $i$-th line of the matrix $K_{l,\alpha} \in \mathbb{R}^{l\times l}$ is equal to $2^{-i+1}\left(C_{i-1}^{i-1}\ C_{i-1}^{i-2}\ \ldots\ C_{i-1}^0\ 0\ \ldots\ 0\right)$, $C_i^\alpha \stackrel{\text{def}}{=} \frac{i!}{\alpha!(i-\alpha)!}$.*

*Proof.* $K_{l,\alpha}$ defined in the statement is such that $\forall v \in \mathbb{C}$, $\left(\frac{v+\bar{v}}{2}\right)^{[l]} = \sum_{\alpha=0}^{l-1} \bar{v}^\alpha K_{l,\alpha} v^{[l]}$. Thus,

$$\begin{aligned}\tilde{M}_l &\left(\left(\frac{z_m+\bar{z}_m}{2}\right)^{[l]} \otimes \cdots \otimes \left(\frac{z_1+\bar{z}_1}{2}\right)^{[l]} \otimes I_n\right) \\ &= \sum_{0\le\alpha_i\le l-1} \bar{z}_1^{\alpha_1} \ldots \bar{z}_m^{\alpha_m} \tilde{M}_l (K_{l,\alpha_m} \otimes \cdots \otimes K_{l,\alpha_1})(z_m^{[l]} \otimes \cdots \otimes z_1^{[l]} \otimes I_n)\ .\end{aligned}$$

The conclusion then follows from the fact that $\forall v \in \mathbb{C}$, $v^\alpha = v^\alpha v^{[1]} = J_{\alpha,1,l-1} v^{[l]}$, so $v^{\alpha *} = v^{[l]*} J_{\alpha,1,l-1}^T$.

### A.2 Products of Polynomial Matrices

Solving the LMIs in Theorems 1 and 2 necessitates to be able to express the coefficient matrix $R_{k+l}$ of $R(z)$ defined in (9), given the coefficient matrices $P_l, N_l$ of $P(z)$, $N(z)$. This in turn necessitates to express the coefficient matrix of a product of matrix-valued polynomials, as function of the coefficient matrices of the factors. This is the goal of Lemma 2.

**Lemma 2.** *Let $l, l' \in \mathbb{N}$, and $M(z), M'(z)$ with coefficient matrices $M_l \in \mathbb{R}^{l^m p \times l^m n}$, $M'_{l'} \in \mathbb{R}^{l'^m n \times l'^m q}$. Then, $M''(z)$ has coefficient matrix $M''_{l''}$, where $l'' = l + l' - 1$ and $M''_{l''} \stackrel{\text{def}}{=} \sum_{\substack{0 \le \alpha_i \le l-1, 0 \le \alpha'_i \le l'-1 \\ 1 \le i \le m}} (J_{\alpha'_m,l,l'-1} \otimes \cdots \otimes J_{\alpha'_1,l,l'-1} \otimes I_p)^T M_l (J_{\alpha_m,1,l-1} \otimes \cdots \otimes J_{\alpha_1,1,l-1} \otimes I_n)^T (J_{\alpha'_m,1,l'-1} \otimes \cdots \otimes J_{\alpha'_1,1,l'-1} \otimes I_n) M'_{l'} (J_{\alpha_m,l',l-1} \otimes \cdots \otimes J_{\alpha_1,l',l-1} \otimes I_q)$.*

*Proof.* One has, $\forall v \in \mathbb{C}$,

$$v^{[l]} = \sum_{\alpha=0}^{l-1} v^\alpha J^T_{\alpha,1,l-1}, \quad v^{[l]} v^{[l']*} = \sum_{\substack{0 \le \alpha \le l-1, \\ 0 \le \alpha' \le l'-1}} v^\alpha \bar{v}^{\alpha'} J^T_{\alpha,1,l-1} J_{\alpha',1,l'-1} \ ,$$

and the proof is achieved by using the fact that $v^\alpha v^{[l']} = J_{\alpha,l',l-1} v^{[l+l'-1]}$, $\bar{v}^{\alpha'} v^{[l]*} = v^{[l+l'-1]*} J^T_{\alpha',l,l'-1}$.

### A.3 Formulas Attached to the Inversion of the Map $\varphi$

Once the LMI (8) or (10) has been solved successfully (for a given $l$), one has to express explicitly $P(\varphi(\sigma))$ and $N(\varphi(\sigma))$ to obtain the gain $K(\sigma) = N(\varphi(\sigma))^{-1} P(\varphi(\sigma))$, departing from the coefficient matrices $P_l, N_l$ of $P(z), N(z)$. This is done with the help of the following result.

**Lemma 3.** *Let $N(z) \in \mathbb{R}^{p \times n}_M[z, \bar{z}]$ with coefficient matrix $N_l \in \mathbb{R}^{p \times n, l^m}_M$. Then,*

$$N(\varphi(\sigma)) = \sum_{\substack{0 \le \alpha_i, \alpha'_i \le l-1 \\ i=1,\ldots,m}} p_{\alpha_1 - \alpha'_1}(\sigma_1) \ldots p_{\alpha_m - \alpha'_m}(\sigma_m)$$
$$\times (J_{\alpha'_m,1,l-1} \otimes \cdots \otimes J_{\alpha'_1,1,l-1} \otimes I_p) N_l (J_{\alpha_m,1,l-1} \otimes \cdots \otimes J_{\alpha_1,1,l-1} \otimes I_n)^T \ ,$$

*where by definition, the polynomials $p_\alpha$ are such that, for any $\phi \in \mathbb{R}$, $\cos(\alpha\phi) = p_\alpha(\cos\phi)$.*

The coefficients of the $p_\alpha$ are easily found, allowing effective use of the previous result. For example, $\cos 2\phi = 2\cos^2\phi - 1$, $\cos 3\phi = 4\cos^3\phi - 3\cos\phi$, so $p_0(\sigma) = 1$, $p_1(\sigma) = \sigma$, $p_2(\sigma) = 2\sigma^2 - 1$, $p_3(\sigma) = 4\sigma^3 - 3\sigma$, and so on.

Forming, from the maps $p_\alpha$, the matrices $T_{l,|\alpha|} \in \mathbb{R}^{1 \times l}$ such that $\forall \alpha \in \{-(l-1), \ldots, 0, \ldots, l-1\}$, $\forall \phi \in \mathbb{R}$, $\cos(\alpha\phi) = T_{l,|\alpha|}(\cos\phi)^{[l]}$, the formula in Lemma 3 writes under matrix form as in (13), with $\tilde{M}_l$ replaced by $\sum_{\substack{0 \le \alpha_i, \alpha'_i \le l-1 \\ i=1,\ldots,m}} (J_{\alpha'_m,1,l-1} \otimes \cdots \otimes J_{\alpha'_1,1,l-1} \otimes I_p) N_l (J^T_{\alpha_m,1,l-1} T_{l,|\alpha_m - \alpha'_m|} \otimes \cdots \otimes J^T_{\alpha_1,1,l-1} T_{l,|\alpha_1 - \alpha'_1|} \otimes I_n)$.

*Proof.* As a direct consequence of the definition, $N(z)$ is equal to

$$\sum_{\substack{0\leq\alpha_i,\alpha_i'\leq l-1\\ i=1,\dots,m}} z_1^{\alpha_1}\bar z_1^{\alpha_1'}\dots z_m^{\alpha_m}\bar z_m^{\alpha_m'}(J_{\alpha_m',1,l-1}\otimes\dots\otimes J_{\alpha_1',1,l-1}\otimes I_p)N_l \\ \times(J_{\alpha_m,1,l-1}\otimes\dots\otimes J_{\alpha_1,1,l-1}\otimes I_n)^T.$$

Taking into account the fact that $|z_i|=1$, $i=1,\dots,m$ and that $N_l\in\mathbb{R}_M^{p\times n,l^m}$, the previous expression is equal to

$$\sum_{\substack{0\leq\alpha_i,\alpha_i'\leq l-1,\alpha_1=\alpha_1'\\ i=1,\dots,m}} z_2^{\alpha_2}\bar z_2^{\alpha_2'}\dots z_m^{\alpha_m}\bar z_m^{\alpha_m'}(J_{\alpha_m',1,l-1}\otimes\dots\otimes J_{\alpha_1',1,l-1}\otimes I_p)N_l \\ \times(J_{\alpha_m,1,l-1}\otimes\dots\otimes J_{\alpha_1,1,l-1}\otimes I_n)^T+ \\ \sum_{\substack{0\leq\alpha_i,\alpha_i'\leq l-1\\ \alpha_1<\alpha_1',\ i=1,\dots,m}} (z_1^{\alpha_1'-\alpha_1}+\bar z_1^{\alpha_1'-\alpha_1})z_2^{\alpha_2}\bar z_2^{\alpha_2'}\dots z_m^{\alpha_m}\bar z_m^{\alpha_m'}(J_{\alpha_m',1,l-1}\otimes\dots\otimes J_{\alpha_1',1,l-1}\otimes I_p)N_l \\ \times(J_{\alpha_m,1,l-1}\otimes\dots\otimes J_{\alpha_1,1,l-1}\otimes I_n)^T.$$

Introducing the functions $p_i$ as defined in the statement, this is also equal to

$$\sum_{\substack{0\leq\alpha_i,\alpha_i'\leq l-1\\ i=1,\dots,m}} p_{\alpha_1-\alpha_1'}(\sigma_1)z_2^{\alpha_2}\bar z_2^{\alpha_2'}\dots z_m^{\alpha_m}\bar z_m^{\alpha_m'}(J_{\alpha_m',1,l-1}\otimes\dots\otimes J_{\alpha_1',1,l-1}\otimes I_p)N_l \\ \times(J_{\alpha_m,1,l-1}\otimes\dots\otimes J_{\alpha_1,1,l-1}\otimes I_n)^T,$$

because $\sigma_1=\operatorname{Re} z_1$. The result follows by induction on $m$.

### A.4 Differentiation of Polynomial Matrices

Lemma 4 below permits to express the coefficient matrix of the terms $\frac{\partial P(\sigma)}{\partial\sigma_i}|_{\sigma=\frac{z+\bar z}{2}}$ in Property II as function of the coefficient matrix of $P(z)$. Notice that the formula therein provides directly the derivatives as a polynomial of degree $k+l-1$ (instead of $l-2$), ready to be added to the term $A(\sigma)P(\sigma)+P(\sigma)A(\sigma)^T+B(\sigma)N(\sigma)+N(\sigma)^TB(\sigma)^T$ in the matrix inequality in Property II, which has precisely the same degree.

**Lemma 4.** *Let* $M(\sigma)\overset{\text{def}}{=}M_l(\sigma_m^{[l]}\otimes\dots\otimes\sigma_1^{[l]}\otimes I_n)$. *Then, for any nonnegative integer* $k$, $\frac{\partial M(\sigma)}{\partial\sigma_i}=\hat M_{k+l,i}(\sigma_m^{[k+l]}\otimes\dots\otimes\sigma_1^{[k+l]}\otimes I_n)$, *with* $\hat M_{k+l,i}\overset{\text{def}}{=}M_l(\hat J_{k,l}^{(m-i)\otimes}\otimes L_l\hat J_{k,l}\otimes\hat J_{k,l}^{(i-1)\otimes}\otimes I_n)$.

*Proof.* Indeed, $\frac{\partial M(\sigma)}{\partial\sigma_i}=M_l(\sigma_m^{[l]}\otimes\dots\otimes\frac{\partial\sigma_i^{[l]}}{\partial\sigma_i}\otimes\sigma_{i-1}^{[l]}\otimes\dots\otimes\sigma_1^{[l]}\otimes I_n)=M_l(I_l^{(m-i)\otimes}\otimes L_l\otimes I_l^{(i-1)\otimes}\otimes I_n)(\sigma_m^{[l]}\otimes\dots\otimes\sigma_1^{[l]}\otimes I_n)=\hat M_{k+l,i}(\sigma_m^{[k+l]}\otimes\dots\otimes\sigma_1^{[k+l]}\otimes I_n)$.

## References


1. P.-A. Bliman (2004). A convex approach to robust stability for linear systems with uncertain scalar parameters, SIAM J. on Control and Optimization, 42(6):2016–2042
2. P.-A. Bliman (2004). An existence result for polynomial solutions of parameter-dependent LMIs, Systems and Control Letters, 51(3-4):165–169.
3. P.-A. Bliman (2004). On robust semidefinite programming. Proc. International Symposium on Mathematical Theory of Networks and Systems (MTNS) Leuven, Belgium.
4. S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan (1994). Linear matrix inequalities in system and control theory. Studies in Applied Mathematics vol. 15, SIAM, Philadelphia.
5. D.J. Leith, W.E. Leithead (2000). Survey of gain-scheduling analysis and design, Int. J. Control, 73(11):1001–1025.
6. G.A. Leonov, D.V. Ponomarenko, V.B. Smirnova (1996). Frequency-domain Methods for Nonlinear Analysis. Theory and Applications. World Scientific Publishing Co.
7. S. Lim, J.P. How (2002). Analysis of linear parameter-varying systems using a non-smooth dissipative systems framework. Int. J. Robust Nonlinear Control. 12:1067–1092.
8. I. Masubuchi (1998). Spline-type solutions to parameter-dependent LMIs. Proc. IEEE Conf. on Decision and Control, Tampa, Florida.
9. I. Masubuchi (1999). An exact solution to parameter-dependent convex differential inequalities. Proc. European Control Conference, Karlsruhe, Germany.
10. I. Masubuchi, T. Akiyama, M. Saeki (2003). Synthesis of Output Feedback Gain-Scheduling Controllers Based on Descriptor LPV System Representation Proc. IEEE Conf. on Decision and Control, Maui, Hawaii.
11. A. Rantzer (1996). On the Kalman-Yakubovich-Popov lemma. Syst. Contr. Lett. 28(1):7–10.
12. W.J. Rugh, J.S. Shamma (2000). Research on gain scheduling. Automatica, 36:1401–1425.
13. G. Szegö, R.E. Kalman (1963). Sur la stabilité absolue d'un système d'équations aux différences finies. Comp. Rend. Acad. Sci., 257(2):338–390.
14. F. Wu (2001). A generalized LPV system analysis and control synthesis framework, Int. J. Control, 74(7):745–759.

# On the Equivalence of Algebraic Approaches to the Minimization of Forms on the Simplex

Etienne de Klerk[1], Monique Laurent[2⋆], and Pablo Parrilo[3]

[1] University of Waterloo, edeklerk@math.uwaterloo.ca
[2] CWI, Amsterdam, M.Laurent@cwi.nl
[3] Swiss Federal Institute of Technology, Zürich, parrilo@control.ee.ethz.ch

We consider the problem of minimizing a form on the standard simplex [equivalently, the problem of minimizing an even form on the unit sphere]. Two converging hierarchies of approximations for this problem can be constructed, that are based, respectively, on results by Schmüdgen-Putinar and by Pólya about representations of positive polynomials in terms of sums of squares. We show that the two approaches yield, in fact, the same approximations.

## 1 Introduction

### 1.1 Representations of positive forms on the simplex

We consider the problem of minimizing a form (i.e., homogeneous polynomial) $p$ of degree $d$ on the standard simplex; that is, the problem of computing

$$p_{\min} := \min\ p(x)\ \text{ s.t. } x \in \Delta := \left\{ x \in \mathbb{R}^n_+ \mid \sum_{i=1}^n x_i = 1 \right\}. \tag{1}$$

The polynomial

$$\tilde{p}(x) := p(x_1^2, \ldots, x_n^2)$$

is an even form of degree $2d$ and problem (1) can be reformulated as the problem of minimizing $\tilde{p}$ on the unit sphere:

$$p_{\min} = \min\ \tilde{p}(x)\ \text{ s.t. } x \in S := \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^2 = 1 \right\}. \tag{2}$$

Equivalently, this is the problem of finding the maximum scalar $t$ for which

$$\tilde{p}(x) - t \geq 0\ \forall x \in S;\ \text{equivalently, } \tilde{p}(x) - t\|x\|^{2d} \geq 0\ \forall x \in \mathbb{R}^n. \tag{3}$$

⋆Supported by the Netherlands Organization for Scientific Research grant NWO 639.032.203.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 121–132, 2005.

Here, $\|x\|^2 = \sum_{i=1}^n x_i^2$. Hence, lower bounds for the optimum value can be obtained by replacing the condition (3) by some stronger conditions. Instances of such stronger conditions are given below, for any integer $r \geq 0$:

$$\left(\tilde{p}(x) - t\|x\|^{2d}\right) \|x\|^{2r} \in \mathbb{R}^+_{ev}[X] \tag{4}$$

$$(\tilde{p}(x) - t\|x\|^{2d})\|x\|^{2r} \in \Sigma^2 \tag{5}$$

$$\tilde{p}(x) - t \in \mathbb{R}^+_{ev,2(r+d)}[X] + (1 - \|x\|^2)\mathbb{R}[X] \tag{6}$$

$$\tilde{p}(x) - t \in \Sigma^2_{2(r+d)} + (1 - \|x\|^2)\mathbb{R}[X] \tag{7}$$

Here, $\mathbb{R}[X]$ denotes the set of polynomials in the $n$ variables $x_1, .., x_n$, $\mathbb{R}^+_{ev}[X]$ is the set of even polynomials with nonnegative coefficients, $\Sigma^2$ is the set of polynomials that are sums of squares, and a subscript $2(r+d)$ indicates the bound $2(r+d)$ on the degree. (See section 1.2 for definitions and notation.)

Note that, in (4), one could replace $\mathbb{R}^+_{ev}[X]$ by $\mathbb{R}^+[X]$, since the polynomial is even by construction.

Condition (4) can be equivalently reformulated in terms of the initial polynomial $p$ as

$$\left(p(x) - t\left(\sum_{i=1}^n x_i\right)^d\right)\left(\sum_{i=1}^n x_i\right)^r \in \mathbb{R}^+[X]. \tag{8}$$

One can also reformulate condition (5) in terms of the original polynomial $p$, using the following result of Zuluaga et al. [16].

**Proposition 1 (Zualaga et al. [16]).** *Let $p$ be a form of degree $d$ and $\tilde{p}(x) := p(x_1^2, \ldots, x_n^2)$ the associated even form. Then,*

$$\tilde{p} \in \Sigma^2 \Longleftrightarrow p(x) = \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I| \equiv d \bmod 2}} \left(\prod_{i \in I} x_i\right) p_I, \quad \textit{where } p_I \in \Sigma^2 \textit{ and } p_I \textit{ is a form of degree } d - |I|$$

The following implications obviously hold:

$$(4) \Longrightarrow (5) \Longrightarrow (3), \quad (6) \Longrightarrow (7) \Longrightarrow (3).$$

Each of the conditions (4)-(7) permits to formulate a hierarchy of lower bounds for $p_{\min}$ depending on $r$. For instance, the (linear) bound:

$$p_L^{(r)} := \max\ t \text{ s.t. (4) (or (8)) holds}, \tag{9}$$

and the (semidefinite) bound:

$$p^{(r)} := \max\ t \text{ s.t. (5) holds}. \tag{10}$$

Obviously,

$$p_L^{(r)} \leq p_L^{(r+1)},\ p^{(r)} \leq p^{(r+1)},\ p_L^{(r)} \leq p^{(r)} \leq p_{\min}. \tag{11}$$

Asymptotic convergence of the bounds $p_L^{(r)}$ to $p_{\min}$ as $r$ goes to infinity, follows from the following theorem of Pólya about representations of positive forms on the simplex. .

**Theorem 1 (Pólya [10]).** *Let $p$ be a form which is positive on the standard simplex $\Delta = \{x \in \mathbb{R}^n \mid \sum_i x_i = 1\}$. Then there exists an $r \in \mathbb{N}$ such that*

$$p(x)\left(\sum_{i-1}^{n} x_i\right)^r \in \mathbb{R}^+[X].$$

Two other hierarchies of lower bounds can be defined analogously, using (6) and (7), and they satisfy the analogue of (11). Their asymptotic convergence to $p_{\min}$ follows from the following theorem of Schmüdgen (or its refinement by Putinar) about representations of positive polynomials on compact semi-algebraic sets.

**Theorem 2.** *Let $F$ be a semi-algebraic set of the form:*

$$F = \{x \in \mathbb{R}^n \mid p_1(x) \geq 0, \ldots, p_k(x) \geq 0\},\ \textit{where } p_1, \ldots, p_k \in \mathbb{R}[X].$$

*(i)* **(Schmüdgen [15])** *If $F$ is compact, then every polynomial which is positive on $F$ belongs to* $\displaystyle\sum_{I \subseteq \{1,\ldots,k\}} \left(\prod_{i \in I} p_i\right) \Sigma^2$.

*(ii)* **(Putinar [13])** *Assume that $F$ is compact and that there exists a polynomial $p_0 \in \Sigma^2 + p_1\Sigma^2 + \ldots + p_k\Sigma^2$ for which the set $\{x \mid p_0(x) \geq 0\}$ is compact. Then every polynomial which is positive on $F$ belongs to $\Sigma^2 + p_1\Sigma^2 + \ldots + p_k\Sigma^2$.*

**Corollary 1.** *Every polynomial which is positive on the unit sphere belongs to $\Sigma^2 + (1 - \sum_{i=1}^n x_i^2)\mathbb{R}[X]$.*

This idea of constructing hierarchies of bounds for optimization over semi-algebraic sets, based on real algebraic results about representations of positive polynomials, has been explored by several authors.

In particular, Pólya's result led Parrilo [8, 9] to introduce hierarchies of conic relaxations for the cone of copositive matrices. These relaxations were used by De Klerk and Pasechnik [6] for approximating the stable set problem in graphs, and by Bomze and De Klerk [1] for constructing a PTAS for the

minimization of degree 2 forms on the simplex. Hierarchies of conic relaxations were introduced, more generally, for the cone of positive semidefinite forms, in particular, by Faybusovich [2] (who also gives estimations on the quality of the approximations) and by Zuluaga et al. [16]. These relaxations have been used in the recent paper by De Klerk, Laurent and Parrilo [5] for giving a PTAS for the minimization of a form of degree $d$ on the simplex.

On the other hand, Putinar's result led Lasserre [7] to define converging hierarchies of semidefinite bounds for the approximation of polynomials on (special) compact semi-algebraic sets.

The main contribution of this paper is to show that these two approaches, based on Pólya's and Schmüdgen-Putinar's theorems, are in fact equivalent, when applied to the problem of minimizing a form on the standard simplex (or, equivalently, minimizing an even form on the unit sphere). More precisely, we prove the following result in Section 2, showing that the assertions (4) and (6) (resp., (5) and (7)) are equivalent.

**Theorem 3.** *Let $p$ be a form of degree $d$ and let $\tilde{p}(x) := p(x_1^2, \ldots, x_n^2)$ be the associated even form of degree $2d$. For every integer $r \geq 0$, consider the linear bound $p_L^{(r)}$ (defined by (9)) and the semidefinite bound $p^{(r)}$ (defined by (10)) for the minimum value $p_{\min}$ of $p$ over the standard simplex. Then,*

$$p_L^{(r)} \leq p^{(r)} \leq p_{\min},$$

$$\begin{aligned} p_L^{(r)} &= \max\ t \quad s.t. \quad \left(\tilde{p}(x) - t\left(\sum_{i=1}^n x_i^2\right)^d\right)\left(\sum_{i=1}^n x_i^2\right)^r \in \mathbb{R}^+[X] \\ &= \max\ t \quad s.t.\ \tilde{p}(x) - t \in \mathbb{R}^+_{ev,2(r+d)}[X] + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X], \end{aligned} \tag{12}$$

$$\begin{aligned} p^{(r)} &= \max\ t \quad s.t. \quad \left(\tilde{p}(x) - t\left(\sum_{i=1}^n x_i^2\right)^d\right)\left(\sum_{i=1}^n x_i^2\right)^r \in \Sigma^2 \\ &= \max\ t \quad s.t.\ \tilde{p}(x) - t \in \Sigma^2_{2(r+d)} + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X]. \end{aligned} \tag{13}$$

We conclude with a 'negative result' in Section 3, concerning representations of polynomials positive on the unit sphere, namely

$$q \in \Sigma^2 + \left(1 - \sum_{i=1}^n x_i^2\right)\Sigma^2 \Longleftrightarrow q \in \Sigma^2.$$

Compare this to the representation $p \in \Sigma^2 + (1 - \sum_{i=1}^n x_i^2)\mathbb{R}[X]$ in Corollary 1 that holds for any $p$ positive on the unit sphere.

### 1.2 Notation

The following notation will be used throughout the paper.

$\mathbb{R}[x_1, \ldots, x_n]$, also abbreviated as $\mathbb{R}[X]$, is the set of polynomials in $n$ variables. Write $p \in \mathbb{R}[X]$ as $\sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha$, where $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Then, $p_\alpha x^\alpha$ is a *term* of $p$ if $p_\alpha \neq 0$; $|\alpha| := \sum_{i=1}^n \alpha_i$ is the *degree* of the term $p_\alpha x^\alpha$, and the *degree* of $p$ is the maximum degree of its terms. A polynomial $p$ is a *form* if all its terms have the same degree; $p$ is an *even* polynomial if $\alpha_1, \ldots, \alpha_n$ are even for every term $p_\alpha x^\alpha$ of $p$.

$\mathbb{R}_d[X]$ is the set of polynomials with degree $\leq d$; $\mathbb{R}^+[X]$ is the set of polynomials with nonnegative coefficients: $p = \sum_\alpha p_\alpha x^\alpha$ with $p_\alpha \geq 0$ for all $\alpha$; $\mathbb{R}_{ev}[X]$ is the set of even polynomials: $p = \sum_\alpha p_\alpha x^{2\alpha}$. Moreover, $\mathbb{R}_d^+[X] := \mathbb{R}^+[X] \cap \mathbb{R}_d[X]$, $\mathbb{R}_{ev}^+[X] := \mathbb{R}^+[X] \cap \mathbb{R}_{ev}[X]$, $\mathbb{R}_{ev,d}^+[X] := \mathbb{R}_{ev}^+[X] \cap \mathbb{R}_d[X]$.

$\Sigma^2$ is the set of polynomials that can be written as a sum of squares of polynomials: $p = \sum_\ell f_\ell^2$ for some $f_\ell \in \mathbb{R}[X]$, and $\Sigma_d^2 := \Sigma^2 \cap \mathbb{R}_d[X]$. Obviously, $\mathbb{R}_{ev}^+[X] \subseteq \Sigma^2$.

## 2 Pólya's and Putinar's Theorems Give the Same Bounds for Optimization on the Simplex

We prove here a slightly more general version of Theorem 3, which holds for forms of even degree. We begin with some preliminary results.

**Proposition 2.** *Let $q$ be a form of even degree $2d \geq 2$. The following assertions are equivalent:*

$$q(x) \left( \sum_{i=1}^n x_i^2 \right)^r \in \mathcal{P} \tag{14}$$

$$q \in \mathcal{P} + \left(1 - \|x\|^2\right) \mathbb{R}[X] \tag{15}$$

*where $\mathcal{P}$ stands for $\mathbb{R}_{ev,2(r+d)}^+[X]$ or $\Sigma_{2(r+d)}^2$.*

*Proof.* Suppose first that (14) holds. Then, the polynomial

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

belongs to $\mathcal{P}$ and

$$f(x) = q(x) \left( 1 - 1 + \sum_{i=1}^n x_i^2 \right)^r = q(x) + \sum_{s=1}^r \binom{r}{s} q(x) \left( \sum_{i=1}^n x_i^2 - 1 \right)^s,$$

which implies that

$$q(x) = f(x) + \left(1 - \|x\|^2\right) \sum_{s=1}^{r} \binom{r}{s} q(x) \left( \sum_{i=1}^{n} x_i^2 - 1 \right)^{s-1}$$

and, thus, (15) holds.

Suppose now that (15) holds; that is,

$$q(x) = s(x) + (1 - \|x\|^2) r(x)$$

where $s \in \mathcal{P}$ and $r \in \mathbb{R}[X]$. Then, $q\left(\frac{x}{\|x\|}\right) = s\left(\frac{x}{\|x\|}\right)$ and, thus,

$$q(x)\|x\|^{2r} = s\left(\frac{x}{\|x\|}\right) \|x\|^{2(r+d)} \quad \text{for all } x \in \mathbb{R}^n \setminus \{0\}. \tag{16}$$

In what follows, we show that

$$f(x) := s\left(\frac{x}{\|x\|}\right) \|x\|^{2(r+d)}$$

is a polynomial belonging to $\mathcal{P}$. This implies that the polynomial $q(x)\|x\|^{2r}$ coincides with $f(x)$ (by continuity) and, thus, belongs to $\mathcal{P}$, which shows that (14) holds.

Suppose first that $\mathcal{P} = \mathbb{R}^+_{ev,2(r+d)}[X]$. Then, $s(x) = \sum_{|\alpha| \le r+d} s_\alpha x^{2\alpha}$, with all $s_\alpha \ge 0$. Therefore, $f(x) = \sum_{|\alpha| \le r+d} s_\alpha x^{2\alpha} \|x\|^{2(r+d-|\alpha|)}$, which is an even polynomial with nonnegative coefficients and, thus, belongs to $\mathcal{P}$.

Suppose now that $\mathcal{P} = \Sigma^2_{2(r+d)}$. We begin with observing that one can assume that each term of $s$ has an even degree. To see it, write $s = s_0 + s_1$, where each term of $s_0$ (resp., of $s_1$) has even (resp., odd) degree. Then, $s_0(-x) = s_0(x)$ and $s_1(-x) = -s_1(x)$ for all $x$. As $q$ is a form of even degree, $q(-x) = q(x)$ for all $x$. In view of (16), this implies that $s(-x) = s(x)$ for all $x$ with $\|x\| = 1$. Therefore, $s_1(x) = 0$ and, thus, $s(x) = s_0(x)$ for all $x$ with $\|x\| = 1$. Hence, one can replace $s$ by $s_0$ in the definition of $f$.

As $s \in \Sigma^2_{2(r+d)}$, write

$$s = \sum_\ell (s_\ell)^2, \;\; s_\ell = u_\ell + v_\ell$$

where $s_\ell$ are polynomials of degree $\le r + d$, $u_\ell$ consists of the terms of $s_\ell$ whose degree has the same parity as $r + d$, and $v_\ell := s_\ell - u_\ell$. Thus,

$$s = \sum_\ell (u_\ell)^2 + (v_\ell)^2 + 2 \sum_\ell u_\ell v_\ell.$$

As each term of $s$, $(u_\ell)^2$, and $(v_\ell)^2$ has even degree, while each term of $u_\ell v_\ell$ has odd degree, we deduce that $\sum_\ell u_\ell v_\ell = 0$, implying that $s = \sum_\ell (u_\ell)^2 + (v_\ell)^2$. Therefore,

$$f(x) = s\left(\frac{x}{\|x\|}\right)\|x\|^{2(r+d)} = \sum_{\ell}\left(u_\ell\left(\frac{x}{\|x\|}\right)\|x\|^{r+d}\right)^2 + \left(v_\ell\left(\frac{x}{\|x\|}\right)\|x\|^{r+d}\right)^2.$$

Observe now that $u_\ell\left(\frac{x}{\|x\|}\right)\|x\|^{r+d} = \varphi_\ell(x)$ and $v_\ell\left(\frac{x}{\|x\|}\right)\|x\|^{r+d} = \|x\|\psi_\ell(x)$ where $\varphi_\ell$ and $\psi_\ell$ are polynomials in $x$. Indeed, say, $u_\ell(x) = \sum_\alpha u_{\ell,\alpha}x^\alpha$. Then, $u_\ell\left(\frac{x}{\|x\|}\right)\|x\|^{r+d}$ is equal to $\sum_\alpha u_{\ell,\alpha}x^\alpha\|x\|^{r+d-|\alpha|}$, which is a polynomial in $x$ since all $r+d-|\alpha|$ are even integers. Analogously for $v_\ell$. This shows that

$$f(x) = \sum_{\ell}\varphi_\ell(x)^2 + \sum_{\ell}\psi_\ell(x)^2\Big(\sum_{i=1}^n x_i^2\Big)$$

belongs to $\mathcal{P}$, thus concluding the proof. ∎

**Lemma 1.** *Let $q$ be a form of even degree $2d$ and let $t$ be a real number. The following assertions are equivalent:*

$$q(x) - t\|x\|^{2d} \in \mathcal{P} + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X] \tag{17}$$

$$q(x) - t \in \mathcal{P} + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X], \tag{18}$$

*where $\mathcal{P}$ stands for $\mathbb{R}^+_{ev,2(r+d)}[X]$ or $\Sigma^2_{2(r+d)}$.*

*Proof.* If (17) holds, then $q(x) - t\|x\|^{2d} = s + \left(1 - \sum_{i=1}^n x_i^2\right)r$, where $s \in \mathcal{P}$ and $r \in \mathbb{R}[X]$. Therefore, $q(x) - t = s + \left(1 - \sum_{i=1}^n x_i^2\right)r + t\left(\|x\|^{2d} - 1\right)$. Now, $\|x\|^{2d} - 1 = \left(\sum_{i=1}^n x_i^2\right)^d - 1 = (\sum_{i=1}^n x_i^2 - 1)u(x)$, for some polynomial $u$. Therefore, (18) holds.

Conversely, if (18) holds, then $q(x) - t = s + \left(1 - \sum_{i=1}^n x_i^2\right)r$, where $s \in \mathcal{P}$ and $r \in \mathbb{R}[X]$. Then, $q(x) - t\|x\|^{2d} = s + \left(1 - \sum_{i=1}^n x_i^2\right)r - t\left(\|x\|^{2d} - 1\right)$ and, thus, (17) holds. ∎

**Theorem 4.** *Let $q$ be a form of even degree $2d$, $q_{\min}$ the minimum of $q(x)$ over the unit sphere, and $r \geq 0$ an integer. Then,*

$$q_L^{(r)} \leq q^{(r)} \leq q_{\min}, \quad \textit{where}$$

$$\begin{aligned} q_L^{(r)} &:= \max\ t \quad s.t. \quad \left(q(x) - t\left(\sum_{i=1}^n x_i^2\right)^d\right)\left(\sum_{i=1}^n x_i^2\right)^r \in \mathbb{R}^+_{ev}[X] \\ &= \max\ t \quad s.t.\ q(x) - t \in \mathbb{R}^+_{ev,2(r+d)}[X] + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X], \end{aligned} \tag{19}$$

$$\begin{aligned} q^{(r)} := \max\ t\ \ s.t.\ \left(q(x) - t\left(\sum_{i=1}^n x_i^2\right)^d\right)\left(\sum_{i=1}^n x_i^2\right)^r \in \Sigma^2 \\ = \max\ t\ \ s.t.\ q(x) - t \in \Sigma^2_{2(r+d)} + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X]. \end{aligned} \tag{20}$$

*Proof.* Follows directly from Proposition 2 (applied to the form $q(x) - t\|x\|^{2d}$) and from Lemma 1. ∎

Therefore, Theorem 3 follows from Theorem 4 applied to the (even) form $q(x) := \tilde{p}(x)$.

We have formulated in Theorem 4 two bounds for the minimum of a form $q$ of even degree on the unit sphere: a linear bound $q_L^{(r)}$ and a semidefinite bound $q^{(r)}$ using, respectively, representations in terms of even polynomials and sums of squares of polynomials. At that point, one should point out that the hierarchy of linear bounds is interesting *only* when $q$ is an *even* form. Indeed, if the form $q$ is not even, then $q_L^{(r)} = -\infty$ for all $r \geq 0$; this follows from the following facts.

**Lemma 2.** *A polynomial $p \in \mathbb{R}[X]$ is even if and only if*

$$p(x_1, \ldots, x_n) = p(-x_1, x_2, \ldots, x_n) = \ldots = p(x_1, \ldots, x_{n-1}, -x_n). \tag{21}$$

*Proof.* Necessity is obvious. Conversely, assume that (21) holds; we show that $p$ is even. For this, let $p_1$ be the sum of the even terms of $p$ and set $q := p - p_1$. Then, $q = \sum_\alpha q_\alpha x^\alpha$ where $\alpha$ has some odd component whenever $q_\alpha \neq 0$. As $p_1$ is an even form, it satisfies (21) and thus $q$ too satisfies (21). We show that $q = 0$, which implies that $p = p_1$ is even. For this, write $q = q_1 + q_2$, where $q_1 := \sum_{\alpha|\alpha_1 \text{ odd}} q_\alpha x^\alpha$. Then, $q(x) = q(-x_1, x_2, \ldots, x_n)$, $q_1(-x_1, x_2, \ldots, x_n) = -q_1(x)$, $q_2(-x_1, x_2, \ldots, x_n) = q_2(x)$; hence,

$$q_1(x) + q_2(x) = q_1(-x_1, x_2, \ldots, x_n) + q_2(-x_1, x_2, \ldots, x_n) = -q_1(x) + q_2(x),$$

which implies that $q_1(x) = 0$. From this follows that $q_\alpha = 0$ whenever $\alpha_1$ is odd. The same reasoning applied to the other coordinates shows that all $q_\alpha$ are equal to 0. ∎

**Corollary 2.** *Given $p \in \mathbb{R}[X]$, the polynomial $p(x)(\sum_{i=1}^n x_i^2)^r$ is even for some $r \geq 0$ if and only if $p$ is even.* ∎

## 3 A Negative Result

Let us now turn to the question of existence of a stronger type of decomposition. Let $q$ be a form of even degree $2d$ which is positive on the unit sphere.

Then, $q(x) > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$. In particular, $q$ is positive on the unit ball $F := \{x \in \mathbb{R}^n \mid 1 - \sum_{i=1}^n x_i^2 \geq 0\}$ except at the origin where it is zero. One may wonder whether an extension of Putinar's result might still hold, permitting to conclude that

$$q \in \Sigma^2 + \left(1 - \sum_{i=1}^n x_i^2\right) \Sigma^2. \tag{22}$$

Scheiderer [14] has recently investigated such extensions of Putinar's result (see Corollary 3.17 in [14]).

**Proposition 3 (Example 3.18 in [14]).** *Let $p \in \mathbb{R}[X]$ be a polynomial for which the level set*

$$K := \{x \in \mathbb{R}^n \mid p(x) \geq 0\}$$

*is compact. Let $q \in \mathbb{R}[X]$ be nonnegative on $K$. Assume that the following conditions hold:*

1. *$q$ has only finitely many zeros in $K$, each lying in the interior of $K$.*
2. *the Hessian $\nabla^2 q$ is positive definite at each of these zeroes.*

*Then $q \in \Sigma^2 + p\Sigma^2$.*

Unfortunately, in the case where $K$ is the unit ball and $q$ a positive semidefinite form of degree at least 4, this theorem does not apply (since the Hessian of $q$ is zero at the origin). In fact, one can show that in this case such a decomposition (22) exists *only* when $q$ itself is a sum of squares.

**Proposition 4.** *Let $q$ be a form of degree $2d$. Then,*

$$q \in \Sigma^2 + \left(1 - \sum_{i=1}^n x_i^2\right) \Sigma^2 \Longleftrightarrow q \in \Sigma^2.$$

*Proof.* The 'if' part being trivial, we prove the 'only if' part. Assume that $q = f + (1 - \sum_{i=1}^n x_i^2)g$, where $f, g \in \Sigma^2$; we show that $q \in \Sigma^2$. Write $f = \sum_\ell f_\ell^2$ and $g = \sum_k g_k^2$. Let $s \geq 0$ be the largest integer for which each term of $f_\ell,\ g_k$ has degree $\geq s$; that is, $f_\ell(x) = \sum_{|\alpha| \geq s} f_{\ell,\alpha} x^\alpha$, $g_k(x) = \sum_{|\alpha| \geq s} g_{k,\alpha} x^\alpha$ for all $\ell, k$ and at least one of the polynomials $f_\ell, g_k$ has a term of degree $s$. Define $f'_\ell$ as the sum of the terms of degree $s$ in $f_\ell$ and $f''_\ell := f_\ell - f'_\ell$; then,

$$f'_\ell(x) = \sum_{|\alpha|=s} f_{\ell,\alpha} x^\alpha,\ f''_\ell(x) = \sum_{|\alpha| \geq s+1} f_{\ell,\alpha} x^\alpha.$$

Analogously, define

$$g'_k(x) := \sum_{|\alpha|=s} g_{k,\alpha} x^\alpha, \; g''_k(x) := \sum_{|\alpha|\geq s+1} g_{k,\alpha} x^\alpha.$$

We have that

$$q = q_1 + q_2, \quad \text{where } q_1 := \sum_\ell (f'_\ell)^2 + \sum_k (g'_k)^2, \quad \text{and}$$

$$q_2 := 2\sum_\ell f'_\ell f''_\ell + 2\sum_k g'_k g''_k + \sum_\ell (f''_\ell)^2 + \sum_k (g''_k)^2 - \left(\sum_{i=1}^n x_i^2\right) g.$$

Therefore, $q_1$ is a (nonzero) form of degree $2s$, while each term of $q_2$ has degree $\geq 2s+1$. If $s \leq d-1$, then $q$ is a form of degree $2d \geq 2s+2$, which implies that $q_1 = 0$, a contradiction. Hence, $s \geq d$ and, in fact, $s = d$. From this follows that $q_2 = 0$ and, thus, $q = q_1$ is a sum of squares. ∎

## 4 Conclusion

We conclude with some comments on the computational implications of Theorem 4 where we showed that

$$\begin{aligned} q^{(r)} &:= \max \; t \;\; \text{s.t.} \;\; \left(q(x) - t\left(\sum_{i=1}^n x_i^2\right)^d\right)\left(\sum_{i=1}^n x_i^2\right)^r \in \Sigma^2 \\ &= \max \; t \;\; \text{s.t.} \; q(x) - t \in \Sigma^2_{2(r+d)} + \left(1 - \sum_{i=1}^n x_i^2\right)\mathbb{R}[X]. \end{aligned}$$

The first representation of $q^{(r)}$ corresponds to various relaxations introduced in the literature for different special cases of the problem

$$q_{\min} = \min \; q(x) \;\; \text{s.t.} \; x \in S := \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^2 = 1 \right\}, \tag{23}$$

by

1. De Klerk and Pasechnik [6] for obtaining the stability number of a graph;
2. Parrilo [9], Bomze and De Klerk [1], Faybusovich [2], and De Klerk, Laurent and Parrilo [5] for minimization of forms on the simplex.

The difficulty with these approaches up to now was that — once an exact relaxation was obtained — it was not clear how to extract a globally optimal solution of problem (23).

The second representation of $q^{(r)}$ in Theorem 4 corresponds exactly to the dual form of the SDP relaxation obtained by applying the general methodology introduced by Lasserre [7] to problem (23).

The approach of Lasserre [7] has now been implemented in the software package Gloptipoly [3] by Henrion and Lasserre.

The authors have also described sufficient conditions for the relaxation of order $r$ to be exact, and have implemented an algorithm for extracting an optimal solution if it is known that the relaxation of order $r$ is exact. The extraction procedure only involves linear algebra on the primal optimal solution of the relaxation; see [4] for details.

Theorem 4 therefore shows how to apply the solution extraction procedure implemented in Gloptipoly to the relaxations studied by De Klerk and Pasechnik [6], Parrilo [9], Bomze and De Klerk [1] and Faybusovich [2].

## References


1. I. Bomze and E. de Klerk (2002). Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. Global Optimization, 24(2):163–185.
2. L. Faybusovich (2003). Global optimization of homogeneous polynomials on the simplex and on the sphere. In C. Floudas and P. Pardalos (Editors). Frontiers in Global Optimization, 109–121. Kluwer Academic Publishers.
3. D. Henrion and J. Lasserre (2003). GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi. ACM Transactions on Mathematical Software, 29.
4. D. Henrion and J. Lasserre (2004). Detecting global optimality and extracting solutions in Gloptipoly. Chapter III.5 in this volume.
5. E. de Klerk and M. Laurent and P.A. Parrilo (2004). A PTAS for the minimization of polynomials of fixed degree over the simplex. Preprint.
6. E. de Klerk and D.V. Pasechnik (2002). Approximation of the stability number of a graph via copositive programming. SIAM Journal on Optimization, 12:875–892.
7. J. Lasserre (2001). Global optimization with polynomials and the problem of moments. SIAM J.Optim. 11(3):796–817.
8. P.A. Parrilo (2000). Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology.
9. P.A. Parrilo (2003). Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming Ser. B, 96:293–320.
10. G. Pólya (1974). Collected Papers, volume 2, pages 309–313. MIT Press, Cambridge, Mass., London.
11. V. Powers and B. Reznick (2001). A new bound for Pólya's theorem with applications to polynomials positive on polyhedra. Journal of Pure and Applied Algebra, 164:221–229.
12. A. Prestel and C.N. Delzell (2001). Positive Polynomials - From Hilbert's 17th Problem to Real Algebra. Springer, Berlin.

13. M. Putinar (1993). Positive polynomials on compact semi-algebraic sets. Ind. Univ. Math. J., 42:969–984.
14. C. Scheiderer (2003). Sums of squares on real algebraic curves. Preprint, to appear in Mathematische Zeitschrift.
15. K. Schmüdgen (1991). The $K$-moment problem for compact semi-algebraic sets. Mathematische Annalen, 289:203–206.
16. L.F. Zuluaga, J.C. Vera, and J.F. Peña (2003). LMI approximations for cones of positive semidefinite forms. Preprint, available on the E-print site Optimization on Line at `www.optimization-online.org/DB_HTML/2003/05/652.html`

# A Moment Approach to Analyze Zeros of Triangular Polynomial Sets

Jean B. Lasserre

LAAS-CNRS, 7 Avenue du Colonel Roche,
31077 Toulouse, France.
lasserre@laas.fr

## 1 Introduction

Consider an ideal $I := \langle g_1, \ldots, g_n \rangle \subset \mathbb{R}[x_1, \ldots, x_n]$ generated by polynomials $\{g_i\}_{i=1}^n \subset \mathbb{R}[x_1, \ldots, x_n]$. Let us call $\mathbf{G} := \{g_1, \ldots, g_n\}$ a *polynomial set* and let a term ordering of monomials with $x_1 < x_2 \ldots < x_n$ be given.

We assume that the system of polynomials equations $\{g_i(x) = 0,\ i = 1, \ldots, n\}$ is in the following triangular form,

$$g_i(x) \ = \ p_i(x_1, \ldots, x_{i-1}) x_i^{r_i} + h_i(x_1, \ldots, x_i) \quad i = 1, \ldots, n \tag{1}$$

by which we mean the following :

- (i) $x_i$ is the main variable and $p_i(x_1, \ldots, x_{i-1}) x_i^{r_i}$ is the leading term of $g_i$.

- (ii) for every $i = 2, \ldots, n$, every zero in $\mathbb{C}^n$ of the polynomial system $\mathbf{G}_{i-1} := \{g_1, \ldots g_{i-1}\}$ is *not* a zero of the leading coefficient $\mathrm{ini}(g_i) := p_i(x_1, \ldots, x_{i-1})$ of $g_i$.

The set $\mathbf{G}$ is called a *triangular set*. From (i)-(ii), it follows that $I$ is a zero-dimensional ideal. Conversely, any zero-dimensional ideal can be represented by a finite union of specific triangular sets (see e.g. Aubry *et al.* [1], Lazard [10]). For various definitions (and results) related to *triangular sets* (e.g. due to Kalkbrener, Lazard, Wu) the interested reader is referred to Lazard [10], Wang [6] and the many references therein; see also Aubry [2] and Maza [11] for a comparison of symbolic algorithms related to triangular sets.

For instance, there are symbolic algorithms that, given $I$ as input, generate a finite set of triangular systems in the specific form $g_i(x) = x_i - f_i(x_1)$ for all $i = 2, \ldots, n$. Triangular sets in the latter form are particularly interesting to develop efficient symbolic algorithms for counting and computing real zeros of polynomials sets (see e.g. Becker and Wörmann [4] and the recent work of Rouillier [12]).

We here show that a triangular polynomial set $\mathbf{G}$ as in (1) has also several advantages from a *numerical point of view*. Indeed, it also permits to

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 133–150, 2005.

define *multivariate Newton sums*, the multivariate analogue of Newton sums for univariate polynomials (which can be used for counting real zeros as in Gantmacher [7, Chap. 15, p. 200]). Namely, we show that :

(a) With a triangular system $\mathbf{G}$ as in (1) we may associate *moment* matrices $M_p(y)$ depending on the (known) multivariate Newton sums of $\mathbf{G}$ (to be defined later) and on an unknown vector $y$. The condition $M_p(y) \succeq 0$ for some specific $p = r_0$ (meaning $M_p(y)$ positive semidefinite) defines a unique solution $y^*$, the vector of all moments (up to order $2p$) of a probability measure $\mu^*$ supported on *all* the zeros of $\mathbf{G}$ in $\mathbb{C}^n$. As a consequence, a polynomial of degree less than $2p$ is in $\sqrt{I}$ if and only if its vector $f$ of coefficients satisfies the linear system of equations $M_p(y^*)f = 0$.

(b) Moreover, given a set

$$\mathbf{K} := \{z \in \mathbb{C}^n \,|\, w_j(z_1, \ldots, z_n, \overline{z_1}, \ldots \overline{z_n}) \geq 0,\ j = 1, \ldots m\} \subset \mathbb{C}^n,$$

defined by some polynomials $\{w_i\}$ in $\mathbb{C}[z, \overline{z}]$ (which can be viewed as a semi-algebraic set in $\mathbb{R}^{2n}$), one may also check whether the zero set of $\mathbf{G}$ is contained in $\mathbf{K}$ by solving a convex *semidefinite program* for which efficient software packages are now available. The necessary and sufficient conditions state that the system of LMI (Linear Matrix Inequalities)

$$M_{r_0}(y) \succeq 0; \quad M_{r_0}(w_i y) \succeq 0 \quad i = 1, \ldots, m,$$

for some appropriate *moment matrix* $M_{r_0}(y)$ and *localizing matrices* $M_{r_0}(w_i y)$ (depending on the Newton sums of $\mathbf{G}$) must have a solution, which is then unique, i.e. $y = y^*$ with $y^*$ as in (a). In fact, it suffices to solve the single inequality $M_{r_0}(y) \succeq 0$ which yields the unique solution $y^*$, and then check *afterwards* whether $M_{r_0}(w_i y^*) \succeq 0$, for all $i = 1, \ldots, m$. For an introduction to semidefinite programming, the interested reader is referred to Vandenberghe and Boyd [14].

The basic technique that we use relies on a deep result of Curto and Fialkow [5] for the $\mathbf{K}$-moment problem.

## 2 Notation, Definitions and Preliminary Results

Some of the material in this section is from Curto and Fialkow [5]. Let $\mathcal{P}_r$ be the space of polynomials in $\mathbb{C}[z_1, \ldots, z_n, \overline{z}_1, \ldots, \overline{z}_n]$ (in short $\mathbb{C}[z, \overline{z}]$) of degree at most $r \in \mathbb{N}$. Now, following notation as in Curto and Fialkow [5], a polynomial $\theta \in \mathbb{C}[z, \overline{z}]$ is written

$$\theta(z, \overline{z}) = \sum_{\alpha\beta} \theta_{\alpha\beta} \overline{z}^{\alpha} z^{\beta} = \sum_{\alpha,\beta} \theta_{\alpha\beta} \overline{z}_1^{\alpha_1} \cdots \overline{z}_n^{\alpha_n} z_1^{\beta_1} \cdots z_n^{\beta_n},$$

in the usual basis of monomials (e.g. ordered lexicographically)

$$1, z_1, \ldots z_n, \overline{z}_1, \ldots, \overline{z_n}, z_1^2, z_1 z_2, \ldots \tag{2}$$

We here identify $\theta \in \mathbb{C}[z, \overline{z}]$ with its vector of coefficients $\theta := \{\theta_{\alpha\beta}\}$ in the basis (2).

Given an infinite sequence $\{y_{\alpha\beta}\}$ indexed in the basis (2), we also define the linear functional on $\mathbb{C}[z, \overline{z}]$

$$\theta \;\mapsto\; \Lambda(\theta) \;:=\; \sum_{\alpha,\beta} \theta_{\alpha\beta} y_{\alpha\beta} \;=\; \sum_{\alpha,\beta} \theta_{\alpha\beta} y_{\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n}.$$

### 2.1 The Moment Matrix

Given $p \in \mathbb{N}$ and an infinite sequence $\{y_{\alpha\beta}\}$ let $M_p(y)$ be the unique square matrix such that

$$\langle M_p(y) f, h \rangle \;=\; \Lambda(f\overline{h}) \quad \forall f, h \in \mathcal{P}_p$$

(see e.g. Curto and Fialkow [5, p. 3]).

To fix ideas, in the two-dimensional case, the moment matrix $M_1(y)$ is given by

$$M_1(y) \;=\; \begin{bmatrix} 1 & y_{0010} & y_{0001} & y_{1000} & y_{0100} \\ y_{1000} & y_{1010} & y_{1001} & y_{2000} & y_{1100} \\ y_{0100} & y_{0110} & y_{0101} & y_{1100} & y_{0200} \\ y_{0010} & y_{0020} & y_{0011} & y_{1010} & y_{0110} \\ y_{0001} & y_{0011} & y_{0002} & y_{1001} & y_{0101} \end{bmatrix}.$$

Thus, the entry of the moment matrix $M_p(y)$ corresponding to column $\overline{z}^\alpha z^\beta$ and row $\overline{z}^\eta z^\gamma$ is $y_{\alpha+\gamma,\beta+\eta}$, and if $y$ is the moment vector of a measure $\mu$ on $\mathbb{C}^n$, then

$$\langle M_p(y) f, f \rangle \;=\; \Lambda(|f|^2) \;=\; \int |f|^2 \, d\mu \;\geq\; 0 \quad \forall f \in \mathcal{P}_p, \tag{3}$$

which shows that $M_p(y)$ is *positive semidefinite* (denoted $M_p(y) \succeq 0$).

### 2.2 Localizing Matrices

Let $\{y_{\alpha\beta}\}$ be an infinite sequence and let $\theta \in \mathbb{C}[z, \overline{z}]$. Define the *localizing* matrix $M_p(\theta y)$ to be the unique square matrix such that

$$\langle M_p(\theta y) f, g \rangle \;=\; \Lambda(\theta f \overline{g}) \quad \forall f, g \in \mathcal{P}_p. \tag{4}$$

Thus, if $\theta(z, \overline{z}) = \sum_{\alpha\beta} \theta_{\alpha\beta} \overline{z}^\alpha z^\beta$ and $M_p(y)(i, j) = y_{\gamma\eta}$ then

$$M_p(\theta y)(i, j) \;=\; \sum_{\alpha\beta} \theta_{\alpha\beta} y_{\alpha+\gamma,\beta+\eta}. \tag{5}$$

For instance, with $z \mapsto \theta(z, \overline{z}) := 1 - \overline{z}_1 z_1$, $M_1(\theta y)$ reads

$$\begin{bmatrix} 1-y_{1010} & y_{0010}-y_{1020} & y_{0001}-y_{1011} & y_{1000}-y_{2010} & y_{0100}-y_{1110} \\ y_{1000}-y_{2010} & y_{1010}-y_{2020} & y_{1001}-y_{2011} & y_{2000}-y_{3021} & y_{1100}-y_{2110} \\ y_{0100}-y_{1110} & y_{0110}-y_{2020} & y_{0101}-y_{1111} & y_{1100}-y_{2110} & y_{0200}-y_{1210} \\ y_{0010}-y_{1020} & y_{0020}-y_{1030} & y_{0011}-y_{1021} & y_{1010}-y_{2020} & y_{0110}-y_{1120} \\ y_{0001}-y_{1011} & y_{0011}-y_{1021} & y_{0002}-y_{1012} & y_{1001}-y_{2011} & y_{0101}-y_{1111} \end{bmatrix}.$$

It follows that if $y$ is the moment vector of some measure $\mu$ on $\mathbb{C}^n$, supported on the set $\{z \in \mathbb{C}^n \,|\, \theta(z,\overline{z}) \geq 0\}$, we then have

$$\langle M_p(\theta y)f, f\rangle \;=\; \Lambda(\theta|f|^2) \;=\; \int \theta|f|^2\, d\mu \;\geq\; 0 \quad \forall f \in \mathcal{P}_p, \tag{6}$$

so that $M_p(\theta y) \succeq 0$.

### 2.3 Multivariate Newton Sums

With $x_1 < x_2, \ldots < x_n$ and given a fixed term ordering of monomials, consider a triangular polynomial system $\mathbf{G} = \{g_1, \ldots, g_n\}$ as in (1), that is,

$$g_i(x) \;=\; p_i(x_1, \ldots x_{i-1})\, x_i^{r_i} + h_i(x_1, \ldots, x_i) \;=\; 0 \quad \forall i = 1, \ldots n \tag{7}$$

(with $p_1 \in \mathbb{R}$), and the $p_i$'s are such that for all $i = 2, 3, \ldots n$,

$$g_k(z) \;=\; 0 \;\forall k = 1, \ldots, i-1 \Rightarrow\; p_i(z) \;\neq\; 0. \tag{8}$$

For each $i = 1, \ldots, n$, $p_i(x_1, \ldots, x_{i-1})x_i^{r_i}$ is the leading term of $g_i$. In the terminology used in e.g. Wang [6, Definitions 2.1], $\mathbf{G}$ is a *triangular set*.

In view of the assumption on the $g_i$'s, it follows that $\mathbf{G}$ has exactly $s := \prod_{i=1}^n r_i$ zeros $\{z(i)\}_{i=1}^s$ in $\mathbb{C}^n$ (counting their multiplicity) so that $I = \langle g_1, \ldots, g_n\rangle$ is a zero-dimensional ideal and the affine variety $V_{\mathbb{C}}(I) \subset \mathbb{C}^n$ is a finite set of cardinality $s_{\mathbf{G}} \leq s$.

For every $\alpha \in \mathbb{N}^n$ define $s_\alpha$ to be the real number

$$s_\alpha \;:=\; s^{-1} \sum_{i=1}^{s} z(i)^\alpha \;=\; \sum_{i=1}^{s} z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_n^{\alpha_n}(i) \tag{9}$$

which we call the (normalized) $\alpha$-Newton sum of $\mathbf{G}$ by analogy with the Newton sums of a univariate polynomial (see e.g. Gantmacher [7, p. 199]).

**Remark 1** Note that the Newton sums $s_\alpha$ *depend* on $\mathbf{G}$ and not only on the zeros $\{z(i)\}$ because we take into account the possible multiplicities.

**Proposition 1.** *Let the $g_i$'s be as in (7)-(8) and $s_\alpha$ be as in (9). Then each $s_\alpha$ is a rational fraction in the coefficients of the $g_i$'s and can be computed recursively.*

For a proof see §5.1.

**Example 2** Consider the elementary example with $\mathbf{G} := \{x_1^2+1, x_1x_2^2+x_2+1\}$. Then, $s_{i0}$ is just the usual (normalized) $i$-Newton sum of $x_1 \mapsto x_1^2+1$. And for instance, it follows that $s_{01} = 0$, $s_{02} = 0$. Similarly, $s_{11} = -1/2$, $s_{21} = 0$, $s_{22} = 1/2$, etc ...

Interestingly, given a polynomial $t \in \mathbb{R}[x_1, \ldots, x_n]$, Rouillier [12, §3] also defines *extended Newton sums* of what he calls a *multi-ensemble* associated with a set of points of $\mathbb{C}^n$. He then uses these extended Newton sums to obtain a certain triangular representation of zero-dimensional ideals.

## 3 Main Result

In this section we assume that we are given a polynomial set $\mathbf{G} := \{g_1, \ldots, g_n\}$ in the triangular form (7)-(8).

### 3.1 The Associated Moment Matrix

The idea in this section is to build up the moment matrices (defined in §2.1) associated with a particular measure $\mu^*$ on $\mathbb{C}^n$ whose support is on *all* the zeros of the polynomial set $\mathbf{G}$. That is, let $\{z(i)\}$ be the collection of $s := \prod_{j=1}^n r_j$ zeros in $\mathbb{C}^n$ of $\mathbf{G}$ (counting their multiplicity) and let $\mu^*$ to be the probability measure on $\mathbb{C}^n$ defined by

$$\mu^* := s^{-1} \sum_{i=1}^{s} \delta_{z(i)}. \tag{10}$$

where $\delta_z$ stands for the Dirac measure at the point $z \in \mathbb{C}^n$.

By definition of $\mu^*$, its moments $\int z^\alpha d\mu^*$ are just the normalized $\alpha$-Newton sums (9). Indeed,

$$s_\alpha := \int z^\alpha \, d\mu^* = s^{-1} \sum_{i=1}^{s} z(i)^\alpha. \tag{11}$$

If we write

$$y^*_{\alpha\beta} := \int \overline{z}^\alpha z^\beta \, d\mu^* \quad \alpha, \beta \in \mathbb{N}^n, \tag{12}$$

we have

$$s_\alpha = y^*_{\alpha 0} = y^*_{0\alpha}; \quad y^*_{\alpha\beta} = y^*_{\beta\alpha} \quad \alpha, \beta \in \mathbb{N}^n. \tag{13}$$

### 3.2 Construction of the Moment Matrix of $\mu^*$

With $\mu^*$ as in (10) let $\{s_\alpha, y^*_{\alpha\beta}\}$ defined in (11)-(12) be the infinite sequence of all its moments.

We then call $M_p(\mu^*)$ the moment matrix associated with $\mu^*$, that is, in $M_p(y)$ we replace the entries $y_{0\alpha}$ or $y_{\alpha 0}$ by $s_\alpha$ and the other entries $y_{\alpha\beta}$ by $y^*_{\alpha\beta}$. By Proposition 1, the entries $s_\alpha$ are known and rational fractions of the coefficients of the polynomials $g_i$'s. They can be computed numerically or symbolically. On the other hand, moments $y^*_{\alpha\beta}$ do not have a closed form expression in terms of the coefficients of polynomials $g_i$'s.

Therefore, we next introduce a moment matrix $M_p(\mu^*, y)$ obtained from $M_p(\mu^*)$ by replacing the (unknown) entries $y^*_{\alpha\beta}$ by variables $y_{\alpha\beta}$ and look for conditions on this matrix $M_p(\mu^*, y)$ to be exactly $M_p(\mu^*)$. For instance, in the two dimensional case, the moment matrix $M_1(\mu^*, y)$ reads

$$M_1(\mu^*, y) = \begin{bmatrix} 1 & s_{10} & s_{01} & s_{10} & s_{01} \\ s_{10} & y_{1010} & y_{1001} & s_{20} & s_{11} \\ s_{01} & y_{0110} & y_{0101} & s_{11} & s_{02} \\ s_{10} & s_{20} & s_{11} & y_{1010} & y_{0110} \\ s_{01} & s_{11} & s_{02} & y_{1001} & y_{0101} \end{bmatrix}.$$

(with $s_\alpha = y_{\alpha 0} = y_{0\alpha}$). Moreover, from the definition of $\mu^*$, we may impose $M_p(\mu^*, y)$ to be symmetric for all $p \in \mathbb{N}$, because $y^*_{\alpha\beta} = y^*_{\beta\alpha}$ for all $\alpha, \beta \in \mathbb{N}^n$ (see (13)).

As $\mathbf{G}$ is a triangular polynomial system in the form (7)-(8), $I = \langle g_1, \ldots, g_n \rangle$ is a zero-dimensional ideal. Therefore, let $H := \{h_1, \ldots, h_m\}$ be a reduced Gröbner basis of $I$ with respect to (in short, w.r.t.) the term ordering already defined (e.g. the lexicographical ordering $x_1 < x_2 \cdots < x_n$). As $I$ is zero-dimensional, for every $i = 1, \ldots n$, we may label the first $n$ polynomials $h_j$ of $H$ in such a way that $x_i^{r'_i}$ is the leading term of $h_i$ (see e.g. Adams and Loustaunau [3, Theor. 2.2.7]).

**Proposition 2.** *Let* $\mathbf{G}$ *be the triangular polynomial system in (7)-(8) (with some term ordering), and let* $H = \{h_1, \ldots, h_m\}$ *be its reduced Gröbner basis (with* $x_i^{r'_i}$ *the leading term of* $h_i$ *for all* $i = 1, \ldots n$*).*

*Let* $\mu^*$ *be the probability measure defined in (10). For every* $\alpha, \beta \in \mathbb{N}^n$ *let*

$$y^*_{\alpha\beta} := \int \overline{z}^\alpha z^\beta \, d\mu^*. \tag{14}$$

*Then, for every* $\gamma, \eta \in \mathbb{N}^n$, $y^*_{\gamma\eta}$ *is a linear combination of the* $y^*_{\alpha\beta}$*'s with* $\alpha_i, \beta_i < r'_i$ *for all* $i = 1, \ldots n$*, that is,*

$$y^*_{\eta\gamma} = \sum_{\alpha,\beta} u_{\alpha\beta}(\eta, \gamma) y^*_{\alpha\beta} \quad \alpha_i, \beta_i < r'_i \quad \forall i = 1, \ldots, n, \tag{15}$$

*for some scalars* $\{u_{\alpha\beta}(\eta, \gamma)\}$.

*Proof.* Let $H = \{h_1, \ldots, h_m\}$ be the reduced Gröbner basis of $I$ w.r.t. the term ordering, with $x_i^{r_i'}$ the leading term of $h_i$ for all $i = 1, \ldots n \leq m$.

For $\eta, \gamma \in \mathbb{N}^n$, write

$$z^\eta = \sum_{i=1}^m q_i(z)h_i(z) + q_\eta(z); \quad \overline{z}^\gamma = \sum_{i=1}^m v_i(\overline{z})h_i(\overline{z}) + v_\gamma(\overline{z}),$$

for some polynomials $\{q_\eta, q_i\}$ and $\{v_\gamma, v_i\}$ in $\mathbb{R}[x_1, \ldots, n]$, that is, $z^\eta$ (resp. $z^\gamma$) are *reduced* to $q_\eta(z)$ (resp. $v_\gamma(z)$) w.r.t. $H$. Due to the special form of $H$, it follows that the monomials $z^\alpha$ of $q_\eta, v_\gamma$ satisfy $\alpha_i < r_i'$ for all $i = 1, \ldots, n$. Hence,

$$q_\eta(z)v_\gamma(\overline{z}) = \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma)\overline{z}^\alpha z^\beta \quad \alpha_i, \beta_i < r_i' \quad \forall i = 1, \ldots, n,$$

for some scalars $\{u_{\alpha\beta}(\eta,\gamma)\}$. Therefore, from the definition of $\mu^*$,

$$\begin{aligned} y^*_{\gamma\eta} = \int z^\eta \overline{z}^\gamma \, d\mu^* &= \int \left(\sum_{i=1}^m q_i(z)h_i(z) + q_\eta(z)\right)\left(\sum_{i=1}^m v_i(\overline{z})h_i(\overline{z}) + v_\gamma(\overline{z})\right) d\mu^* \\ &= \int q_\eta(z)v_\gamma(\overline{z})\, d\mu^* = \sum_{\alpha\beta} u_{\alpha\beta}(\eta,\gamma) \int \overline{z}^\alpha z^\beta \, d\mu^* \\ &= \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) y^*_{\alpha\beta}, \qquad \alpha_i, \beta_i < r_i' \quad \forall i = 1, \ldots, n \end{aligned}$$

The $y^*_{\alpha\beta}$'s with $\alpha_i, \beta_i < r_i'$, correspond to the *irreducible* monomials $x^\alpha$, $x^\beta$ with respect to the Gröbner basis $H$, which form a basis of $\mathbb{R}[x_1, x_2, \ldots, x_n]/I$ viewed as a vector space over $\mathbb{R}$. In fact, in view of the triangular form (7)-(8), the Gröbner basis $H$ of $I$ w.r.t. to the lexicographical ordering $x_1 < \ldots < x_n$ is such that $r_i' = r_i$ for all $i = 1, \ldots, n$ and $H$ has exactly $n$ terms (Rouillier [13]).

In view of Proposition 2, we may redefine the moment matrix $M_p(\mu^*, y)$ in an equivalent form as follows:

**Definition 3 (Construction of $M_p(\mu^*, y)$)** Let $H = \{h_1, \ldots, h_m\}$ be a reduced Gröbner basis of $I$ w.r.t. to the given term ordering (with $x_i^{r_i'}$ the leading term of $h_i$ for all $i = 1, \ldots, n \leq m$).

The moment matrix $M_p(\mu^*, y)$ is the moment matrix $M_p(y)$ defined in §2.1 and where :

- every entry $y_{\alpha 0}$ or $y_{0\alpha}$ of $M_p(y)$ is replaced with the (known) $\alpha$-Newton sum $s_\alpha$ of $\mathbf{G}$.
- every entry $y_{\gamma\eta}$ in $M_p(y)$ is replaced with the linear combination (15) of $\{y_{\alpha\beta}\}$ with $\alpha_i, \beta_i < r_i'$ for all $i = 1, \ldots, n$.

Thus, in this equivalent formulation, only a finite number of variables $y_{\alpha\beta}$ are involved in $M_p(\mu^*, y)$, all with $\alpha_i, \beta_i < r_i'$ for all $i = 1, \ldots, n$.

**Remark 4** The above definition of $M_p(\mu^*, y)$ *depends* on the reduced Gröbner basis $H$ of $\mathbf{G}$, whereas the entries $s_\alpha$ only depend on the the $g_i$'s.

**Example 5** Let

$$\mathbf{G} := \{x_1^3 + x_1,\ (x_1^2+3)x_2^3 - x_1^2x_2^2 + (x_1^2 - x_1 - 1)x_2 - x_1 + 1\}.$$

Then,

$$H = \{x_1^3 + x_1;\ 6x_2^3 - 3x_1^2x_2^2 + 4x_2x_1^2 - 3x_2x_1 - 2x_2 - x_1^2 - 3x_1 + 2\},$$

and, for instance, denoting "$\rightharpoonup$" the reduction process w.r.t. $H$,

$$z_2^3 \rightharpoonup (3z_1^2z_2^2 - 4z_2z_1^2 + 3z_2z_1 + 2z_2 + z_1^2 + 3z_1 - 2)/6,$$

and as $z_1^4 \rightharpoonup -z_1^2$, we have

$$y_{4003} = (-3y_{2022} + 4y_{2021} - 3y_{2011} - 2y_{2001} - y_{2020} - 3y_{2010} + 2y_{2000})/6,$$

and the latter expression can be substituted for every occurrence of $y_{4003}$.

**Theorem 6.** *Let* $\mathbf{G}$ *be a triangular polynomial system as in (7)-(8) and let* $\{s_\alpha\}$ *be the Newton sums of* $G$ *in (9). Let* $M_p(\mu^*, y)$ *be the moment matrix as in Definition 3, and let* $r_0 := 2\sum_{j=1}^n (r'_j - 1)$. *Then :*

(i) *For all* $p \geq r_0$, $M_p(\mu^*, y) = M_p(\mu^*)$ *if and only if* $M_p(\mu^*, y) \succeq 0$.

(ii) *For all* $p \geq r_0$, $\mathrm{rank}(M_p(\mu^*)) = \mathrm{rank}(M_{r_0}(\mu^*))$, *the number of distinct zeros in* $\mathbb{C}^n$ *of the polynomial system* $\mathbf{G}$.

(iii) *Let* $f \in \mathbb{C}[z, \overline{z}]$ *be of degree less than* $2p$. *All the zeros in* $\mathbb{C}^n$ *of the polynomial system* $\mathbf{G}$ *are zeros of* $f$ *if and only if*

$$M_p(\mu^*)f = 0. \tag{16}$$

*In particular, a polynomial* $f \in \mathbb{R}[x_1, \ldots, x_n]$ *of degree less than* $2p$ *is in* $\sqrt{I}$ *if and only if (16) holds.*

The proof is postponed to §5.2.

**Remark 7** (a) Theorem 6(i) also states that for all $p \geq r_0$, $M_p(\mu^*, y) \succeq 0$ has a unique feasible solution, namely the vector $y^*$ of moments of the probability measure $\mu^*$, truncated up to order $2p$. In particular, solving $M_{r_0}(\mu^*, y) \succeq 0$ provides the vector $y^*$ of all moments of $\mu^*$, truncated up to order $2r_0$. One then gets all the other moments $\{y^*_{\alpha\beta}\}$ by (15) in Proposition 2.

(b) Theorem 6(iii) has an equivalent formulation as follows. Let $f \in \mathbb{C}[z, \overline{z}]$ be of degree at most $2p$ and let $\hat{f}$ be its reduction w.r.t. $H$, the Gröbner basis of $\mathbf{G}$ defined in Proposition 2. Then the condition $M_p(\mu^*)f = 0$ is equivalent to $M_{r_0}\hat{f} = 0$.

(c) Given a reduced Gröbner basis $H$ of $I$, the condition $M_{r_0}(\mu^*, y) \succeq 0$ in Theorem 6(i) is equivalent to the same condition for its submatrix $\widehat{M}_{r_0}(\mu^*, y)$ whose indices of rows and columns in the basis (2) correspond to *independent* monomials $\{z^\alpha\}$ which form a basis of $\mathbb{R}[x_1, \ldots, x_n]/I$, their conjugate $\{\overline{z}^\alpha\}$ and the corresponding monomial products $\overline{z}^\alpha z^\beta$. Indeed, the positive semidefinite condition on the latter is equivalent to the positive semidefinite condition on the former.

**Example 8** Consider the trivial example $\mathbf{G} := \{x^2+1\}$ so that $V_{\mathbb{C}}(I) = \{\pm i\}$. Then the condition $\widehat{M}_{r_0}(\mu^*, y) \succeq 0$ (see Remark 7(b)) reads

$$\widehat{M}_2(\mu^*, y) \;=\; \begin{bmatrix} 1 & 0 & 0 & y_{11} \\ 0 & y_{11} & -1 & 0 \\ 0 & -1 & y_{11} & 0 \\ y_{11} & 0 & 0 & 1 \end{bmatrix} \;\succeq\; 0,$$

which clearly implies $y_{11} = 1 = \int \overline{z} z \, d\mu^*$. Moreover,

$$\operatorname{rank}(M_{r_0}(\mu^*, y)) \;=\; \operatorname{rank}(\widehat{M}_{r_0}(\mu^*, y)) \;=\; 2 = |V_{\mathbb{C}}(I)|.$$

Similarly, let $\mathbf{G} := \{x_1^2 + 1, x_1 x_2 + 1\}$ so that $V_{\mathbb{C}}(I) = \{(i, -i), (-i, i)\}$. We have $r_0 = 2$ and with the lexicographical ordering $x_1 < x_2$, $H := \{x_1^2 + 1, x_2 - x_1\}$ is a reduced Gröbner basis of $I$. Hence, in the moment matrix $M_{r_0}(\mu^*, y)$ every $y_{\alpha_1\alpha_2\beta_1\beta_2}$ is replaced with $y_{\alpha_1+\beta_1,0,\alpha_2+\beta_2,0}$. Moreover, we only need to consider $\alpha_1, \beta_1 < 1$. Therefore, we only need to consider the monomials $\{z_1, z_2, \overline{z_1}, \overline{z_2}, z_1\overline{z_1}\}$ and and in view of Remark 7(b), the (equivalent) condition $\widehat{M}_{r_0}(\mu^*, y) \succeq 0$ reads (denoting $y_{1010} = y$)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & y \\ 0 & y & y & -1 & -1 & 0 \\ 0 & y & y & -1 & -1 & 0 \\ 0 & -1 & -1 & y & y & 0 \\ 0 & -1 & -1 & y & y & 0 \\ y & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \;\succeq\; 0,$$

which implies $y = 1 = \int \overline{z}_1 z_1 \, d\mu^*$. Moreover,

$$\operatorname{rank}(M_{r_0}(\mu^*, y)) \;=\; \operatorname{rank}(\widehat{M}_{r_0}(\mu^*, y)) \;=\; 4 \;=\; |V_{\mathbb{C}}(I)|.$$

### 3.3 Conditions for a Localization of Zeros of G

Let $w_i \in \mathbb{C}[z, \overline{z}], i = 1, \ldots m$, be given polynomials and let $\mathbf{K} \subset \mathbb{C}^n$ be the set defined by

$$\mathbf{K} \;:=\; \{z \in \mathbb{C}^n \,|\, w_i(z, \overline{z}) \;\geq\; 0, \quad i = 1, \ldots, m\} \tag{17}$$

(which can be viewed as a semi-algebraic set in $\mathbb{R}^{2n}$). We now consider the following issue :

*Under what conditions on the coefficients of the polynomials $g_i$'s are all the zeros of the triangular system* $\mathbf{G}$ *contained in* $\mathbf{K}$*?*

Let $M_p(w_i y)$ be the localizing matrices (cf. §2.2) associated with the polynomials $w_i$, for all $i = 1, \ldots, m$. As we did for the moment matrix $M_p(\mu^*, y)$ in Definition 3, we define $M_p(\mu^*, w_i, y)$ to be the localizing matrix $M_p(w_i y)$ where the entries $y_{0\alpha}$ and $y_{\alpha 0}$ are replaced with the $\alpha$-Newton sums $s_\alpha$, and where all the $y_{\eta\gamma}$ are replaced by the linear combinations (15) of the $\{y_{\alpha\beta}\}$ with $\alpha_i, \beta_i < r'_i$ for all $i = 1, \ldots, n$. Accordingly, $M_p(\mu^*, w_i)$ is obtained from $M_p(w_i y)$ by replacing $y$ with $y^*$ as in Proposition 2.

**Theorem 9.** *Let* $\mathbf{G}$ *be the triangular system in (7)-(8) and let* $M_{r_0}(\mu^*, y)$ *be as in Theorem 6. Then, all the zeros of* $\mathbf{G}$ *are in* $\mathbf{K}$ *if and only if*

$$M_{r_0}(\mu^*, w_i) \succeq 0, \qquad i = 1, \ldots, m. \tag{18}$$

*Equivalently, all the zeros of* $\mathbf{G}$ *are in* $\mathbf{K}$ *if and only if the system of linear matrix inequalities*

$$M_{r_0}(\mu^*, y) \succeq 0; \quad M_{r_0}(\mu^*, w_i, y) \succeq 0 \quad i = 1, \ldots, m \tag{19}$$

*has a solution $y$.*

*Proof.* The necessity is obvious. Indeed, assume that all the zeros of $\mathbf{G}$ are in $\mathbf{K}$. Let $\mu^*$ be as in (10) and let $y^* := \{s_\alpha, y^*_{\alpha\beta}\}$ be the infinite sequence of moments of $\mu^*$. Then, of course, $M_p(\mu^*) \succeq 0$ and

$$M_p(\mu^*, w_i) \;=\; M_p(w_i y^*) \succeq 0 \quad i = 1, \ldots, m,$$

for all $p \in \mathbb{N}$, is a necessary condition for $\mu^*$ to have its support in $\mathbf{K}$. Thus, $y := \{y^*_{\alpha\beta}\}$ is a solution of (19).

Conversely, let $y$ be a solution of (19). From Theorem 6(i) $\{s_\alpha, y_{\alpha\beta}\}$ is the moment vector of $\mu^*$, that is, $\{y_{\alpha\beta}\} = \{y^*_{\alpha\beta}\}$ for all $\alpha, \beta$ with $\alpha_i, \beta_i < r'_i$, for all $i = 1, \ldots, n$. Then, all the other $y^*_{\alpha\beta}$ can be obtained from the former by (15). Therefore, and in view of the construction of the localizing matrices $M_p(\mu^*, w_i, y)$, we have

$$M_p(\mu^*, w_i, y) \;=\; M_p(\mu^*, w_i, y^*) \;=\; M_p(\mu^*, w_i).$$

Moreover, using the terminology of Curto and Fialkow [5] (see also the proof of Theorem 6), all the moment matrices $M_p(\mu^*, y) = M_p(\mu^*)$ $(p > r_0)$ are *flat positive extensions* of $M_{r_0}(\mu^*, y) = M_{r_0}(\mu^*)$. As $M_{r_0}(\mu^*, w_i, y) = M_{r_0}(\mu^*, w_i) \succeq 0$, it follows from Theorem 1.6 in Curto and Fialkow [5] that $\mu^*$ has its support contained in $\mathbf{K}$. Hence, as $\mu^*$ is supported on *all* the zeros of $\mathbf{G}$, all the zeros of $\mathbf{G}$ are in $\mathbf{K}$.

## 4 Conclusion

We have considered a system $\mathbf{G}$ of polynomial equations in triangular form and show that several characterizations of the zeros of $\mathbf{G}$ may be obtained from *positive semidefinite* (numerical) conditions on appropriate *moment* and *localizing* matrices. In particular, the triangular form of $\mathbf{G}$ permits to define the analogue for the multivariate case of *Newton sums* in the univariate case.

## 5 Proofs

### 5.1 Proof of Proposition 1

*Proof.* The proof is by induction. In view of the triangular form (7)-(8), the zero set of $\mathbf{G}$ in $\mathbb{C}^n$ (or, equivalently, the variety $V_{\mathbb{C}}(I)$ associated with $I$) consists of $s := \prod_{j=1}^n r_j$ zeros that we label $z(i)$, $i = 1, \ldots, s$, counting their multiplicity.

In addition, still in view of (7)-(8), any particular zero $z(i) \in \mathbb{C}^n$ of $\mathbf{G}$ can be written

$$z(i) \;=\; [z_1(i_1), z_2(i_1, i_2), \ldots, z_n(i_1, \ldots, i_n)],$$

for some multi-index $i_1 \leq r_1, \ldots, i_n \leq r_n$, and where each $z_k(i_1, \ldots, i_k) \in \mathbb{C}$ is a zero of the univariate polynomial $x \mapsto g_k(z_1(i_1), \ldots, z_{k-1}(i_1, \ldots, i_{k-1}), x)$ (where multiplicy is taken into account).

Therefore, for every $\alpha \in \mathbb{N}^n$, the $\alpha$-Newton sum $y_\alpha$ defined in (9) can be written

$$sy_\alpha \;:=\; \sum_{i=1}^{s} z(i)^\alpha \;=\; \sum_{i_1 \leq r_1, \ldots, i_n \leq r_n} z_1(i_1)^{\alpha_1} z_2(i_1, i_2)^{\alpha_2} \cdots z_n(i_1, \ldots, i_n)^{\alpha_n}. \tag{20}$$

Let us make the following induction hypothesis.

$H_k$. *For every* $p, q \in \mathbb{R}[x_1, \ldots, x_k]$

$$\begin{aligned} S_k(p,q) &:= \sum_{i_1, \ldots, i_k} \frac{p(z_1(i_1), \ldots, z_k(i_k))}{q(z_1(i_1), \ldots, z_k(i_k))} \\ &= \sum_{i_1, \ldots, i_k} \frac{\sum_\alpha p_\alpha z_1(i_1)^{\alpha_1} \cdots z_k(i_k)^{\alpha_k}}{\sum_\alpha q_\alpha z_1(i_1)^{\alpha_1} \cdots z_k(i_k)^{\alpha_k}} \end{aligned} \tag{21}$$

*is a rational fraction of coefficients of the polynomials* $g_i$*'s,* $i = 1, \ldots, k$.

Observe that (20) is a particular case of (21) in $H_n$.

We first prove that $H_1$ is true. Let $p, q \in \mathbb{R}[x_1]$ and

$$S(p,q) \;=\; \sum_{j=1}^{r_1} \frac{\sum_k p_k z_1(j)^k}{\sum_k q_k z_1(j)^k},$$

with $\{z_1(j)\}$ being the zeros of $x_1 \mapsto g_1(x_1)$, counting their multiplicity.

Reducing to a common denominator, $S(p,q)$ reads

$$S(p,q) \;=\; \frac{P(z_1(1),\ldots,z_1(r_1))}{Q(z_1(1),\ldots,z_1(r_1))},$$

for some *symmetric* polynomials $P, Q$ of the variables $\{z_1(j)\}$ and whose coefficients are polynomials of coefficients of $p, q$.

Therefore, by the fundamental theorem of symmetric functions, both numerator $P(.)$ and denominator $Q(.)$ are rational fractions of coefficients of $g_1$ (polynomials if $g_1$ is monic). Thus, $H_1$ is true and we can write $S_1(p,q) \;=\; u_{pq}(g_1)/v_{pq}(g_1)$ for some polynomials $u_{pq}, v_{pq}$ of coefficients of $g_1$. The coefficients of $u_{pq}, v_{pq}$ are themselves polynomials of coefficients of the polynomials $p, q$.

Next, assume that $H_j$ is true for all $1 \le j \le k$, that is : for all $j = 1,\ldots,k$ and $p, q \in \mathbb{R}[x_1,\ldots,x_j]$,

$$S_j(p,q) \;=\; u_{pq}(g_1,\ldots,g_j)/v_{pq}(g_1,\ldots,g_j) \tag{22}$$

for some polynomials $u_{pq}, v_{pq}$ of coefficients of the polynomials $g_1,\ldots,g_j$.

We are going to show that $H_{k+1}$ is true. Let $p, q \in \mathbb{R}[x_1,\ldots,x_{k+1}]$ and

$$S_{k+1}(p,q) \;=\; \sum_{i_1,\ldots,i_{k+1}} \frac{\sum_\alpha p_\alpha z_1(i_1)^{\alpha_1}\cdots z_{k+1}(i_1,\ldots,i_{k+1})^{\alpha_{k+1}}}{\sum_\alpha q_\alpha z_1(i_1)^{\alpha_1}\cdots z_{k+1}(i_1,\ldots,i_{k+1})^{\alpha_{k+1}}}.$$

$S_{k+1}(p,q)$ can be rewritten as

$$S_{k+1}(p,q) = \sum_{i_1,\ldots,i_k} \Bigg[ \sum_{j=1}^{r_{k+1}} \frac{\sum_\alpha p_\alpha z_1(i_1)^{\alpha_1}\cdots z_k(i_1,\ldots,i_k)^{\alpha_k} z_{k+1}(i_1,\ldots,i_k,j)^{\alpha_{k+1}}}{\sum_\alpha q_\alpha z_1(i_1)^{\alpha_1}\cdots z_k(i_1,\ldots,i_k)^{\alpha_k} z_{k+1}(i_1,\ldots,i_k,j)^{\alpha_{k+1}}} \Bigg]. \tag{23}$$

In (23), the term

$$A := \left[ \sum_{j=1}^{r_{k+1}} \frac{\sum_\alpha p_\alpha z_1(i_1)^{\alpha_1}\cdots z_k(i_1,\ldots,i_k)^{\alpha_k} z_{k+1}(i_1,\ldots,i_k,j)^{\alpha_{k+1}}}{\sum_\alpha q_\alpha z_1(i_1)^{\alpha_1}\cdots z_k(i_1,\ldots,i_k)^{\alpha_k} z_{k+1}(i_1,\ldots,i_k,j)^{\alpha_{k+1}}} \right]$$

can in turn be written as

$$A \;=\; \sum_{j=1}^{r_{k+1}} \frac{\tilde{p}(z_{k+1}(i_1,\ldots,i_k,j))}{\tilde{q}(z_{k+1}(i_1,\ldots,i_k,j))}, \tag{24}$$

for some univariate polynomials $\tilde{p}, \tilde{q} \in \mathbb{R}[x]$ of the variable $z_{k+1}(i_1,\ldots,i_k,j)$, which is a zero of the univariate polynomial

$$x \mapsto g_{k+1}(z_1(i_1), \ldots, z_k(i_1, \ldots, i_k), x),$$

and whose coefficients are polynomials in the variables $z_1(i_1), z_2(i_1, i_2), \ldots, z_k(i_1, \ldots, i_k)$. In view of $H_1$

$$A = \frac{u_{\tilde{p}\tilde{q}}(g_{k+1})}{v_{\tilde{p}\tilde{q}}(g_{k+1})},$$

for some polynomials $u_{\tilde{p}\tilde{q}}, v_{\tilde{p}\tilde{q}}$ of the coefficients of $g_{k+1}$.

The coefficients of the polynomials $u_{\tilde{p}\tilde{q}}, v_{\tilde{p}\tilde{q}}$ are themselves polynomials of coefficients of $p, q$ and of $z_1(i_1), \ldots, z_k(i_1, \ldots, i_k)$. Hence, substituting for $A$ in (23) we obtain

$$\begin{aligned} S_{k+1}(p,q) &= \sum_{i_1,\ldots,i_k} \frac{\sum_\alpha U_\alpha(g_{k+1}) z_1(i_1)^{\alpha_1} \cdots z_k(i_1,\ldots,i_k)^{\alpha_k}}{\sum_\alpha V_\alpha(g_{k+1}) z_1(i_1)^{\alpha_1} \cdots z_k(i_1,\ldots,i_k)^{\alpha_k}} \\ &= S_k(U(g_{k+1}), V(g_{k+1})) \end{aligned}$$

for some polynomials $U, V \in \mathbb{R}[x_1, \ldots, x_k]$ whose coefficients are polynomials of coefficients of $g_{k+1}$.

We next use the induction hypothesis $H_k$ by which $S_k(U(g_{k+1}), V(g_{k+1}))$ is a rational fraction $f_{UV}(g_1, \ldots, g_k)/h_{UV}(g_1, \ldots, g_k)$ of coefficients of the polynomials $g_1, \ldots, g_k$. As the coefficients of $f_{UV}, h_{UV}$ are themselves rational fractions of coefficients of $g_{k+1}$ we finally obtain that

$$S_{k+1}(p,q) = \frac{u'_{pq}(g_1, \ldots, g_{k+1})}{v'_{pq}(g_1, \ldots, g_{k+1})},$$

that is, a rational fraction of coefficients of the polynomials $g_1, \ldots, g_{k+1}$. Hence $H_{k+1}$ is true, and therefore, the induction hypothesis is true.

Now Proposition 1 follows from $H_n$ and the expression (20) for the $\alpha$-Newton sum $y_\alpha$. That the $\{y_\alpha\}$ can be computed recursively is clear from the above proof of the induction hypothesis $H_k$.

### 5.2 Proof of Theorem 6

*Proof.* (i) Let $p > r_0$ be fixed, arbitrary, and write

$$M_p(\mu^*, y) = \begin{bmatrix} M_{r_0}(\mu^*, y) & | & B \\ - & & - \\ B' & | & C \end{bmatrix}.$$

Consider an arbitrary column $\begin{bmatrix} B(.,j) \\ C(.,j) \end{bmatrix}$. By definition of the moment matrix, $B(1,j)$ is a monomial $z^\gamma \overline{z}^\eta$ for which $\gamma_i > r'_i$ or $\eta_k > r'_k$ for at least one index $i$ or $k$. By Proposition 2

$$z^\gamma \overline{z}^\eta \;=\; \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma)\overline{z}^\alpha z^\beta \quad \alpha_i,\beta_i < r'_i,\ \forall i = 1,\ldots,n, \tag{25}$$

for some scalars $\{u_{\alpha\beta}(\eta,\gamma)\}$, so that, from the construction of $M_p(\mu^*,y)$, we have

$$\begin{aligned} B(1,j) \;=\; y_{\eta\gamma} &= \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) y_{\alpha\beta} \\ &= \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) M_{r_0}(\mu^*,y)(1,j_{\alpha\beta}), \end{aligned}$$

where $j_{\alpha\beta}$ is the index of the column of $M_{r_0}(\mu^*,y)$ corresponding to the monomial $\overline{z}^\alpha z^\beta$. Next, consider an element $B(k,j)$ of the column $B(.,j)$. The element $k$ of $M_p(\mu^*,y)(k,1)$ is a monomial $z^p\overline{z}^q$ and from the definition of $M_p(\mu^*,y)$, we have $B(k,j) = y_{\eta+q,\gamma+p}$. Now, from (25) we have

$$z^p\overline{z}^q z^\gamma \overline{z}^\eta \;=\; \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) z^{\beta+p}\overline{z}^{\alpha+q},$$

which implies

$$y_{\eta+q,\gamma+p} \;=\; \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) y_{\alpha+q,\beta+p},$$

or, equivalently,

$$B(k,j) \;=\; \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) M_{r_0}(k,j_{\alpha\beta}).$$

The same argument holds for $C(.,j)$. Hence,

$$\begin{bmatrix} B \\ C \end{bmatrix}(j) \;=\; \sum_{\alpha,\beta} u_{\alpha\beta}(\eta,\gamma) \begin{bmatrix} M_{r_0}(\mu^*,y) \\ B' \end{bmatrix}(j) \quad \forall j,$$

which, in view of $M_p(\mu^*,y) \succeq 0$, implies that

$$\mathrm{rank}(M_p(\mu^*,y)) \;=\; \mathrm{rank}(M_{r_0}(\mu^*,y)).$$

As $p > r_0$ was arbitrary, and using the terminology of Curto and Fialkow [5], it follows that the matrices $M_p(\mu^*,y)$ are *flat positive extensions* of $M_{r_0}(\mu^*,y)$ for all $p > r_0$. This in turn implies that indeed, the entries of $M_{r_0}(\mu^*,y)$ are moments of some $\mathrm{rank}(M_{r_0}(\mu^*,y))$-atomic probability measure $\mu$.

We next prove that $\mu = \mu^*$, i.e., the condition $M_{r_0}(\mu^*,y) \succeq 0$ determines a unique vector $y = y^*$ that corresponds to the vector of moments of $\mu^*$, up to order $2r_0$.

Given the Gröbner basis $H = \{h_i\}_{i=1}^m$ of $I = \langle g_1,\ldots,g_n\rangle$ (already considered in the proof of Proposition 2), let $\overline{h}_i \in \mathbb{C}[z,\overline{z}]$ be the conjugate polynomial of $h_i$, i.e., $\overline{h}_i(z,\overline{z}) = h_i(\overline{z})$ for all $i = 1,\ldots,m$. We first prove that

$$M_p(h_i y) \;=\; 0; \quad M_p(\overline{h}_i y) = 0, \quad i \;=\; 1,\ldots,m, \quad p \in \mathbb{N}, \tag{26}$$

where $M_p(h_i y)$ (resp. $M_p(\overline{h}y)$) is the *localizing matrix* associated with the polynomials $h_i$ (resp. $\overline{h}_i$).

By Proposition 2, recall that any entry $y_{\eta\gamma}$ of $M_p(\mu^*, y)$ is replaced by a linear combination of the $y_{\alpha\beta}$'s with $\alpha_i, \beta_i < r_i'$ for all $i = 1, \ldots, n$. This linear combination is coming from the *reduction* of the monomials $\{z^\alpha\}_{\alpha\in\mathbb{N}^n}$ with respect to $H$; that is, let us call $J$ the set of indices $\beta$ corresponding to the irreducible monomials $z^\beta$ w.r.t. $H$. Then, the reduction of $z^\alpha$ w.r.t. $H$ yields

$$z^\alpha \;=\; \sum_{i=1}^m q_i(z)h_i(z) + \sum_{\beta\in J} u_\beta(\alpha) z^\beta \quad \text{denoted } z^\alpha \;\rightharpoonup\; \sum_{\beta\in J} u_\beta(\alpha) z^\beta,$$

and similarly,

$$\overline{z}^\alpha \;=\; \sum_{i=1}^m q_i(\overline{z})h_i(\overline{z}) + \sum_{\beta\in J} u_\beta(\alpha) \overline{z}^\beta \quad \text{denoted } \overline{z}^\alpha \;\rightharpoonup\; \sum_{\beta\in J} u_\beta(\alpha) \overline{z}^\beta.$$

From this, we obtain (see the proof of Proposition 2)

$$z^\gamma \overline{z}^\eta \;\rightharpoonup\; \left(\sum_{\beta\in J} u_\beta(\gamma) z^\beta\right)\left(\sum_{\beta\in J} u_\beta(\eta)\overline{z}^\beta\right) \;\rightharpoonup\; \sum_{\alpha,\beta\in J} u_{\alpha\beta}(\eta,\gamma)\overline{z}^\alpha z^\beta, \qquad (27)$$

for some scalars $\{u_{\alpha\beta}(\eta,\gamma)\}$, and thus, the entry $y_{\eta\gamma}$ of $M_p(\mu^*, y)$ is replaced with $\sum_{\alpha,\beta\in J} u_{\alpha\beta}(\eta,\gamma) y_{\alpha\beta}$, or, equivalently,

$$y_{\eta\gamma} - \sum_{\alpha,\beta\in J} u_{\alpha\beta}(\eta,\gamma) y_{\alpha\beta} \;=\; 0. \qquad (28)$$

So let $p \in \mathbb{N}$ be fixed, and consider the entry $M_p(h_i y)(k,l)$ of the localizing matrix $M_p(h_i y)$. Recall that $M_p(y)(k,l) = y_{\phi\zeta}$ for some $\phi, \zeta \in \mathbb{N}^n$, and so, $M_p(h_i y)(k,l)$ is just the expression $\overline{z}^\phi z^\zeta h_i(z)$ where each monomial $\overline{z}^\alpha z^\beta$ is replaced with $y_{\alpha\beta}$; see (5). Next, by definition, $\overline{z}^\phi z^\zeta h_i \rightharpoonup 0$ for all $i = 1, \ldots, m$. Therefore, when $y$ is as in Definition 3 (that is, when (28) holds), writing

$$\overline{z}^\phi\, z^\zeta h_i \;=\; \sum_{\eta,\gamma\in\mathbb{N}^n} v_{\eta\gamma}\overline{z}^\eta z^\gamma \;\rightharpoonup\; 0,$$

and using (27)-(28), yields

$$M_p(h_i y)(k,l) \;=\; \sum_{\alpha,\beta\in J}\left(\sum_{\eta,\gamma\in\mathbb{N}^n} v_{\eta\gamma} u_{\alpha\beta}(\eta,\gamma)\right) y_{\alpha\beta} \;=\; 0.$$

Recall that $p \in \mathbb{N}$, and $k, l$ were arbitrary. Therefore, when $y$ is as in Definition 3, we have $M_p(h_i y) = 0$ (and similarly, $M_p(\overline{h}_i y) = 0$), for all $i = 1, \ldots, m$, and all $p \in \mathbb{N}$. That is, (26) holds.

Hence, let $\mu$ be the $r$-atomic probability measure encountered earlier (with $r := \mathrm{rank}(M_{r_0}(\mu^*, y))$), and let $\{z(k)\}_{k=1}^r \subset \mathbb{C}^n$ be the $r$ distinct points of the support of $\mu$, that is,

$$\mu = \sum_{k=1}^r u_k \delta_{z(k)}, \quad \sum_{k=1}^r u_k = 1; \qquad 0 < u_k, \quad k = 1, \ldots, r,$$

with $\delta_\bullet$ the Dirac measure at $\bullet$.

For every $1 \le i \le r$, let $q_i \in \mathbb{C}[z, \overline{z}]$ be a polynomial that vanishes at all $z(k)$, $k \ne i$, and with $q_i(z(i), \overline{z(i)}) \ne 0$. Let $p \ge \deg q_i$. Then for all $j = 1, \ldots, m$, we have (denoting also $q_i$ as the vector of coefficients of $q_i \in \mathbb{C}[z, \overline{z}]$)

$$0 = \langle q_i, M_p(h_j y) q_i \rangle = \int |q_i(z, \overline{z})|^2 h_j(z)\, \mu(dz) = u_i |q_i(z(i), \overline{z(i)})|^2 h_j(z(i)),$$

and so, $h_j(z_i) = 0$ for all $j = 1, \ldots, m$.

As this is true for all $1 \le i \le r$, it follows that

$$h_j(z(i)) = 0, \quad i = 1, \ldots, r; \quad j = 1, \ldots, m,$$

that is, $\mu$ has its support contained in $\mathbf{G}$. Therefore, with $\{z(i)\}_{i=1}^{s_0}$ being the distinct zeros in $\mathbb{C}^n$ of $\mathbf{G}$,

$$\mu = \sum_{i=1}^{s_0} u_i \delta_{z(i)}, \quad \sum_{i=1}^{s_0} u_i = 1, \quad u_i \ge 0, \quad i = 1, \ldots, n,$$

for some nonnegative scalars $\{u_i\}$, whereas $\mu^* = s^{-1} \sum_{i=1}^s \delta_{z(i)}$ (counting their multiplicity) or $\mu^* = \sum_{i=1}^{s_0} v_i \delta_{z(i)}$ for some nonnegative scalars $\{v_i\}$.

Remember that by definition of the matrices $M_{r_0}(\mu^*)$ and $M_{r_0}(\mu^*, y)$, their entries $\{s_\alpha\}$ (corresponding to the Newton sums) are the same. That is,

$$s_\alpha = \int z^\alpha \, d\mu = \int z^\alpha \, d\mu^*, \qquad \alpha_j \le r_j - 1, \quad j = 1, \ldots, n.$$

Now, we also know that $s_0$ is less than the number of *independent* monomials $\{z^{\beta(j)}\}$ (w.r.t. $H$) which form a basis of $\mathbb{R}[x_1, \ldots, x_n]/I$ (with equality if $I = \sqrt{I}$). Therefore, if $\mu \ne \mu^*$, we have

$$\sum_{i=1}^{s_0} (u_i - v_i) z(i)^{\beta(j)} = 0, \quad j = 1, \ldots s_0, \text{ with } u \ne v,$$

which yields that the square matrix of the above linear system is singular. Hence some linear combination $\{\lambda_j\}$ of its rows vanishes, i.e.,

$$\sum_{j=1}^{s_0} \lambda_j z(k)^{\beta(j)} = 0, \quad \forall k = 1, \ldots, s_0,$$

in contradiction with the linear independence of the $\{z^{\beta(j)}\}$. Hence $u = v$, which in turn implies $\mu = \mu^*$. So it follows that $M_{r_0}(\mu^*, y) \succeq 0$ has only one solution, namely $y = y^*$, the (truncated) vector $y^*$ of moments up to order $2r_0$, of the probability measure $\mu^*$.

Finally, this implies that $s_0 = r = \text{rank}(M_{r_0}(\mu^*, y)) = \text{rank}(M_{r_0}(\mu^*))$ because by Curto and Fialkow [5, Theor. 1.6], the number of atoms of $\mu = \mu^*$ is precisely $\text{rank}(M_{r_0}(\mu^*, y))$. This also proves that $M_{r_0}(\mu^*, y) = M_{r_0}(\mu^*)$ and thus, (i) and (ii).

To prove (iii), consider a polynomial $f \in \mathbb{C}[z, \overline{z}]$ of degree less than $2p$ with coefficient vector in the basis (2) still denoted $f$. It is clear that if $f(z(i)) = 0$ for all $i = 1, \ldots, s_0$ then

$$0 = \int |f|^2 \, d\mu^* = \langle M_p(\mu^*) f, f \rangle,$$

which in turn implies $M_p(\mu^*) f = 0$. Conversely,

$$M_p(\mu^*) f = 0 \Rightarrow 0 = \langle M_p(\mu^*) f, f \rangle = \int |f|^2 \, d\mu^*,$$

which in turn implies $f(z) = 0$, $\mu^*$-a.e.

Finally, let $f \in \mathbb{R}[x_1, \ldots, x_n]$. Recall that $\sqrt{I} = I(V_{\mathbb{C}}(I))$ where $V_{\mathbb{C}}(I) = \{z(i)\}_{i=1}^{s_0}$, that is, $f \in \sqrt{I}$ if and only if $f(z(i)) = 0$ for all $i = 1, \ldots, s_0$. In view of what precedes, $f \in \sqrt{I}$ if and only if $M_p(\mu^*) f = 0$.

## References


1. Aubry P., Lazard D., Moreno Maza M. (1999). On the theories of triangular sets. J. Symb. Comp. 28: 105–124.
2. Aubry P., Moreno Maza M. (1999). Triangular sets for solving polynomial systems: A comparative implementation of four methods. J. Symb. Comput. 28: 125-154.
3. Adams W.W., Loustaunau P. (1994). An Introduction to Gröbner Bases, American Mathematical Society, Providence, Rhode Island.
4. Becker E., Wörmann T. (1996). Radical computations of zero-dimensional idelas and real root counting, Math. Comp. Simul. 42: 561–569.
5. Curto R.E., Fialkow L.A. (2000). The truncated complex $K$-moment problem, Trans. Amer. Math. Soc. 352: 2825-2855.
6. Donming Wang (2000). Computing triangular systems and regular systems, J. Symb. Comp. 30: 221–236.
7. Gantmacher F.R. (1966). Théorie des Matrices. II. Questions spéciales et applications, Dunod, Paris.
8. Lasserre J.B. (2002). Polynomials with all zeros real and in a prescribed interval, J. Alg. Comb. 16: 31–237.
9. Lasserre J.B. (2004) Characterizing polynomials with roots in a semi-algebraic set, IEEE Trans. Aut. Contr. 49: 727–730.

10. Lazard D. (1992) Solving zero-dimensional algebraic systems, J. Symb. Comp. 13: 117–131.
11. Moreno Maza M. (2000). Integration of triangular sets methods in Aldor, Technical report.
12. Rouillier F. (1998). Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux, PhD thesis (in french), INRIA, Metz, France.
13. Rouillier F. (2002). Private communication.
14. Vandenberghe L., Boyd S. (1996). Semidefinite programming, SIAM Review 38: 49-95.

# Polynomials Positive on Unbounded Rectangles

Victoria Powers[1⋆] and Bruce Reznick[2⋆⋆]

[1] Department of Mathematics and Computer Science,
Emory University,
Atlanta, GA 30322
[2] Department of Mathematics
University of Illinois at Urbana-Champaign,
Urbana, IL 61801

## 1 Introduction and Notation

Given a semialgebraic set $K \subseteq \mathbb{R}^N$ determined by a finite set of polynomial inequalities $\{g_1 \geq 0, \ldots, g_k \geq 0\}$, we want to characterize a polynomial $f$ which is positive (or non-negative) on $K$ in terms of sums of squares and the polynomials $g_i$ used to describe $K$. Such a representation of $f$ is an immediate witness to the positivity condition. Theorems about the existence of such representations also have various applications, notably in problems of optimizing polynomial functions on semialgebraic sets.

In case $K$ is compact, Schmüdgen has proved that any polynomial which is positive on $K$ is in the preorder generated by the $g_i$'s, i.e., the set of finite sums of elements of the form $s_e g_1^{e_1} \ldots g_k^{e_k}, e_i \in \{0, 1\}$, where each $s_e$ is a sum of squares of polynomials. Putinar has proved that, under certain conditions, the preorder can be replaced by the quadratic module , which is the set of sums $\{s_0 + s_1 g_1 + \cdots + s_k g_k\}$, where each $s_i$ is a sum of squares . Using this result, Lasserre has developed algorithms for finding the minimum of a polynomial on such compact $K$, which transforms this into a semidefinite programming problem.

What happens when $K$ is not compact? Scheiderer has shown that if $K$ is not compact and $\dim K \geq 3$, then Schmüdgen's characterization can never

⋆This material is based in part on work of this author performed while she was a visiting professor at Universidad Complutense, Madrid, Spain with support from the D.G.I. de España.

⋆⋆This material is based in part upon work of this author, supported by the USAF under DARPA/AFOSR MURI Award F49620-02-1-0325. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of these agencies.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 151–163, 2005.

hold, regardless of the $g_i$'s chosen to describe $K$. Kuhlman and Marshall settled the case where $K$ is not compact and contained in $\mathbb{R}$; the answer here depends on chosing the "right" set of generators for $K$. In this paper we consider some variations on these themes: we look at some canonical non-compact sets in $\mathbb{R}^2$ which are products of intervals and at some stronger and weaker versions of positivity.

We introduce some basic notation. Let $S = \{g_1, \ldots, g_s\}$ denote a finite set of polynomials in $R_n := \mathbb{R}[x_1, \ldots, x_n]$, and let

$$K = K_S = \bigcap_j \{a \in \mathbb{R}^n \mid f_j(a) \geq 0\}.$$

Write $\sum R_n^2$ for the set of finite sums of squares of elements of $R_n$; clearly, any $\sigma \in \sum R_n^2$ takes only non-negative values on $\mathbb{R}^n$. We shall say that an element of $\sum R_n^2$ is *sos*. The *preorder* generated by $S$, denoted $T_S$, is the set of finite sums of the type $\sum \sigma g_1^{\epsilon_1} \ldots g_s^{\epsilon_s}$ where $\sigma$ is sos and $\epsilon_i \in \{0, 1\}$. That is, a typical element of $T_S$ has the shape

$$\sigma_0 + \sum_I \sigma_I \left( \prod_{i \in I} g_I \right),$$

where the sum is taken over all non-empty $I \subseteq \{1, \ldots, s\}$, and each $\sigma_-$ is sos. An important subset of the preorder is the *quadratic module*, $M_S$, which consists of sums in which $\sum_i \epsilon_i \leq 1$ for each summand. That is, a typical element of $M_S$ has the shape

$$\sigma_0 + \sum_{k=1}^{s} \sigma_k g_k.$$

Clearly, $M_S \subseteq T_S$, and if $s \geq 2$, then inclusion is formally strict. However, there can be non-trivial equality. For example, if $S = \{1 - x, 1 + x\}$, then the identity

$$(1 + x)(1 - x) = \left( \frac{(1 - x)^2}{2} \right) (1 + x) + \left( \frac{(1 + x)^2}{2} \right) (1 - x) \tag{1}$$

shows that $(1 + x)(1 - x)$ is already in $M_S$, so $M_S = T_S$ for this case.

**Various notions of positivity.** For $K \subseteq \mathbb{R}^n$ and $f \in R_n$, we write $f \geq 0$ on $K$ if $f(x) \geq 0$ for all $x \in K$ and $f > 0$ on $K$ if $f(x) > 0$ for all $x \in K$. We consider a stronger version of positivity which considers positivity at "points at infinity". (This definition appeared in [13, Ch. 7], in the context of moment and quadrature problems.)

For $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, write $(x, 1)$ for $(x_1, \ldots, x_n, 1) \in \mathbb{R}^{n+1}$, and let

$$x^* := \frac{(x, 1)}{|(x, 1)|} \in S^n \subset \mathbb{R}^{n+1}.$$

Suppose $K \subseteq \mathbb{R}^n$ is a closed set. Let $K^* = \{x^* \mid x \in K\}$ and let $\overline{K^*}$ be the closure of $\{x^* \mid x \in K\}$. For example, if $K = \mathbb{R}^2$, then $K^*$ consists of the Northern Hemisphere, and $\overline{K^*}$ is the Northern Hemisphere plus the equator.

Suppose $p \in R_n$ of degree $d$ and let $p^* \in R_{n+1}$ be the homogenization of $p$, i.e.,

$$p^*(x_1, \ldots, x_{n+1}) := x_{n+1}^d \; p\left(\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}\right).$$

For $x \in \mathbb{R}^n$, let $\Phi_{n,d}(x_1, \ldots, x_n) := (1 + \sum_{i=1}^n x_i^2)^{d/2} = |(x, 1)|^d$. It follows from homogeneity that

$$p(x) = p^*(x, 1) = \Phi_{n,d}(x) p^*(x^*). \tag{2}$$

We say $p$ is *projectively positive on* $K$ if $p^*$ is positive on $\overline{K^*}$, and write $p \gg 0$ on $K$. Clearly, $p \gg 0$ on $K \Rightarrow p > 0$ on $K$. A simple example shows that the converse is false: Let $K = \mathbb{R}^2$ and $p(x, y) = x^2y^2 + 1$, so that $p^*(x, y, z) = x^2y^2 + z^4$. Then $p > 0$ on $K$; but $p^*(1, 0, 0) = 0$, so $p$ is not projectively positive on $K$. Observe that $\overline{K^*}$ is compact, and so if $p \gg 0$ on $K$, then $p^*$ achieves a positive minimum on $\overline{K^*}$.

**Proposition 1.** *Suppose $K \subseteq \mathbb{R}^n$ is closed and $p \in R$ of degree $d$.*

*(i) There exists $c > 0$ so that $p - c\,\Phi_{n,d} \geq 0$ on $K$ iff $p \gg 0$ on $K$.*
*(ii) If $K$ is compact, then $p \gg 0$ on $K$ iff $p > 0$ on $K$.*
*(iii) If $(x_1, \ldots, x_{n+1}) \in \overline{K^*} \setminus K^*$, then $x_{n+1} = 0$.*

*Proof.* By (2), we have $p^* \geq c > 0$ on $\overline{K^*}$ if and only if

$$p(x) = \Phi_{n,d}(x) p^*(x^*) \geq c\,\Phi_{n,d}(x).$$

for $x \in K$, proving (i). For (ii), it suffices to show that $p > 0$ on $K$ implies $p \gg 0$ on $K$. Since $\Phi_{n,d}$ is bounded (say, by $M$) on the compact set $K$, $p \geq c$ on $K$ implies that $p^* \geq c/M$ on $\overline{K^*}$. Finally, suppose

$$(x_1, \ldots, x_{n+1}) = \lim_{N \to \infty} (x_1^{(N)}, \ldots, x_{n+1}^{(N)}),$$

where $(x_1^{(N)}, \ldots, x_{n+1}^{(N)}) \in K^*$ and $x_{n+1} > 0$. Then $x_{n+1}^{(N)} > 0$ for $N \geq N_0$, and for each such $N$,

$$\left(\frac{x_1^{(N)}}{x_{n+1}^{(N)}}, \ldots, \frac{x_n^{(N)}}{x_{n+1}^{(N)}}\right)$$

belongs to $K$. Since $K$ is closed, the limit is in $K$, and by retracing the definition, we see that $(x_1, \ldots, x_{n+1}) \in \overline{K^*}$.

Fix $S$, and let $K = K_S, M = M_S$ and $T = T_S$. We consider six properties of $S$:

$$
\begin{aligned}
&(*) \quad f \gg 0 \text{ on } K \Rightarrow f \in T \\
&(*)_M \quad f \gg 0 \text{ on } K \Rightarrow f \in M \\
&(**) \quad f > 0 \text{ on } K \Rightarrow f \in T \\
&(**)_M \quad f > 0 \text{ on } K \Rightarrow f \in M \\
&(***) \quad f \geq 0 \text{ on } K \Rightarrow f \in T \\
&(***)_M \quad f \geq 0 \text{ on } K \Rightarrow f \in M
\end{aligned}
$$

There is an immediate diagram of implications:

$$
\begin{array}{ccccc}
(***) & \Rightarrow & (**) & \Rightarrow & (*) \\
\Uparrow & & \Uparrow & & \Uparrow \\
(***)_M & \Rightarrow & (**)_M & \Rightarrow & (*)_M
\end{array}
$$

In case $s = 1$, the two rows of properties coalesce; if $K$ is compact, then the last two columns coalesce.

We summarize what is known about these properties: Schmüdgen's Theorem [16] says that if $K$ is compact, then $(**)$ holds, regardless of the choice of $S$. Also, when $K$ is compact, Putinar [12] has shown that $(**)_M$ holds iff $M$ contains a polynomial of the form $r - \sum x_i^2$ for some non-negative $r$.

On the other hand, Scheiderer [15] has shown that if $K$ is not compact and $\dim K \geq 3$, or if $\dim K = 2$ and $K$ contains a two-dimensional cone, then $(**)$ fails. (Observe that $K$ contains a two-dimensional cone iff $\overline{K^*}$ contains an arc on the equator of the unit sphere.) The proof is non-constructive. The case of non-compact semialgebraic subsets of $\mathbb{R}$ has been settled completely by Kuhlmann and Marshall [3]. They show that in this case, $(**)$ and $(***)$ are equivalent and hold iff $S$ contains a specific set of polynomials which generate $S$ (what they call the "natural set of generators"). They also show that $(**)_M$ and $(***)_M$ are equivalent, and only hold in a few special cases.

In general, $(***)$ will not hold, even in the compact case. An easy example is given by $S = \{(1 - x^2)^3\}$ , in which case $K_S = [-1, 1]$ but $1 - x^2 \notin T_S$. See [17] for details on this example.

The authors [10] previously considered two special cases in $\mathbb{R}$, in which $K = [-1, 1]$ or $[0, \infty)$. (It is easily shown via linear changes of variable that the case of closed intervals in $\mathbb{R}$ reduces to one of $\{[-1, 1], [0, \infty), \mathbb{R}\}$.) For each of these intervals, $(***)_M$ has been long known, for the "natural set of generators". Hilbert knew (and saw no need to prove) that if $f(x) \geq 0$ for $x \in \mathbb{R}$, then $f$ is a sum of (two) squares of polynomials; this corresponds precisely to $T_\emptyset$. If we take $S = \{1 + x, 1 - x\}$, so that $K = [-1, 1]$, then Bernstein proved $(**)$ in 1915. On the other hand, if $S = \{1-x^2\}$, then Fekete proved $(***)$ (some time before 1930, the first reference seems to be [8]). The authors showed as Corollary 14 in [10] that if $S$ is given and $K = [0, \infty)$, then

$\epsilon + x \in T_S$ for all $\epsilon$ iff $S$ contains $cx$ for some $c > 0$. In other words, $(**)$ fails for the (non-compact) set $[0, \infty)$ unless the natural generator is included.

In view of the foregoing, this paper considers products of intervals in the plane. There are, up to linear changes and permutation of the variables, six cases of products of closed intervals in the plane, and we take the natural set of generators:

$$\begin{aligned}
K_0 &= [-1,1] \times [-1,1] & S_0 &= \{1-x, 1+x, 1-y, 1+y\};\\
K_1 &= [-1,1] \times [0,\infty) & S_1 &= \{1-x, 1+x, y\};\\
K_2 &= [-1,1] \times (-\infty,\infty) & S_2 &= \{1-x, 1+x\};\\
K_3 &= [0,\infty) \times [0,\infty) & S_3 &= \{x, y\};\\
K_4 &= [0,\infty) \times (-\infty,\infty) & S_4 &= \{x\};\\
K_5 &= (-\infty,\infty) \times (-\infty,\infty) & S_5 &= \emptyset.
\end{aligned}$$

Since $K_0$ is compact, $f > 0$ and $f \gg 0$ are equivalent and $(**)$ holds. By the Putinar result, $(**)_M$ holds in this case as well. Scheiderer recently showed [15] that $(***)$ holds for $K_0$. Thus all but the possibly $(***)_M$ hold hold for $K_0$.

All properties fail for $K_5$, by classical results of Hilbert and Robinson. $K_3$ and $K_4$ contain two-dimensional cones, so Scheiderer's work implies that $(**)$ fails for them; we shall present simple examples in the next section. In fact, we will show that $(*)$ does not hold in these cases. Thus all properties fail for $K_3$ and $K_4$.

Finally, we consider $K_1$ and $K_2$. We show that $(**)_M$ does not hold for $K_1$ and that $(*)$ holds for $K_2$. It is still an open question whether or not $(**)$ holds for $K_1$ or $K_2$.

**Projective positivity and optimization**. Recently, there has been interest in using representation theorems such as those of Schmüdgen and Putinar for developing algorithms for optimizing polynomials on semialgebraic sets. Lasserre [4] [5] describes a method for finding a lower bound for the minimum of a polynomial on a basic closed semialgebraic set and shows that the method produces the exact minimum in the compact case. Marshall [6] shows that in the presence of a certain stability condition, the general problem can be reduced to the compact case, and hence can be handled using Lasserre's method. It turns out that Marshall's stability condition is intimately related to projective positivity.

**Definition 1 (Marshall).** *Suppose $S = \{g_1, \ldots, g_s\} \subseteq R_n$ and $f \in R_n$ is bounded from below on $K_S$. We say $f$ is stably bounded from below on $K_S$ if for any $h \in R_n$ with $\deg h \leq \deg f$, there exists $\epsilon > 0$ so that $f - \epsilon h$ is also bounded from below on $K_S$.*

**Theorem 1 (Marshall).** *Suppose $S$ is given as above and $f$ is stably bounded from below on $K_S$. Then there is a computable $\rho > 0$ so that the minimum of $f$ on $K_S$ occurs on the (compact) semialgebraic set $K_S \cap \{x \mid \rho - ||x||^2 \geq 0\}$.*

We now interpret Proposition 1 in terms of projective positivity.

**Proposition 2.** *Given $S = \{g_1, \ldots, g_s\}, f \subseteq R_n$. Then $f \gg 0$ on $K_S$ implies $f$ is stably bounded from below by 0 on $K_S$.*

*Proof.* By Proposition 1, $f \gg 0$ iff there is $c \in \mathbb{R}^+$ so that $f - c\Phi(n,d)(x) > 0$ on $K_S$. Given $h \in R_n$ with $\deg h = d$, then there is some $N > 0$ and $\epsilon > 0$ such that $\epsilon p(x) < c\Phi(n,d)(x)$ for $||x|| > N$. Then $f - \epsilon p > 0$ on $K_S \cap \{x \mid ||x|| > N\}$ and this implies $f - \epsilon p$ is bounded from below on $K_S$.

Thus for applications to optimization, projective positivity is the "right" notion of positivity to consider. As Marshall remarks in [6]: "In cases where $f$ is not stably bounded from below on $K_S$, any procedure for approximating the minimum of $f$ using floating point computations involving the coefficients is necessarily somewhat suspect."

## 2 The Plane, Half Plane, and Quarter Plane

In the section we consider the semialgebraic sets $K_3$, $K_4$, and $K_5$ with generators $S_3$, $S_4$, and $S_5$. As stated above, it has been shown that $(**)$ holds neither for $K_5$ (Hilbert) nor for $K_3$ and $K_4$ (Scheiderer). In this section, we will construct explicit examples showing that $(*)$ does not hold, which implies $(**)$ does not hold.

First we consider polynomials $f \in R_2 := R = \mathbb{R}[x,y]$ which are nonnegative in the plane and review some results about when they are in $\Sigma R^2$. We shall use the standard terminology that $p$ is *psd* if $p \geq 0$ on $\mathbb{R}^2$ and $p$ is *pd* if $p > 0$ on $\mathbb{R}^2$ In 1888, Hilbert [2] gave a construction of a non-sos polynomial which is psd on $\mathbb{R}^2$. This construction was not explicit, and the first explicit example was found by Motzkin [7] in 1967. R. M. Robinson simplified Hilbert's construction [14]; we will use this example to construct the counterexamples in this section:

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

For $a \geq 0$, let $Q_a(x,y,z) = Q(x,y,z) + a(x^2+y^2+z^2)^3$, then since $Q$ is psd, $Q_a$ is also pd for $a > 0$. It is shown in [14] (and the observation really goes back to [2]) that the cone of sos ternary sextic forms is closed. Since $Q$ does not belong to this cone, it follows that for *some* positive value of $a$, $Q_a$ is not sos. In fact, the methods of [1] can be used to show that $Q_a$ is

pd but not sos for $a \in (0, 1/48)$; we omit the details. Let $q_a(x,y) \in \mathbb{R}[x,y]$ be the dehomogenization of $Q_a$, then for $0 < a < 1/48$, $q_a$ is pd and not sos. As already noted, $\overline{K_5^*}$ is the Northerm Hemisphere plus the equator, and $q_a^* = Q_a$, hence $q_a \gg 0$ on $K_5$ and $q_a$ is not in $T_5$. Thus $(*)$ does not hold for $K_5$.

Note that $Q$ is even in $x$ and so we can consider $f(x,y) = q_a(\sqrt{x}, y)$, so that $f(x^2, y) = q_a(x,y)$. Then $f^*(x^2, y, z) = Q_a(x,y,z)$, hence $f^*(x,y,z) \geq 0$ for $x > 0$. It is easy to see that $\overline{K_4^*}$ is the quarter sphere plus half the equator; thus $f \gg 0$ on $K_4$. But if $f \in T_4$, then there exist sos $\sigma_j$ so that

$$f(x,y) = \sigma_0(x,y) + x\sigma_1(x,y).$$

If we replace $x$ by $x^2$ above, we obtain

$$Q_a(x,y) = f(x^2,y) = \sigma_0(x^2,y) + x^2\sigma_1(x^2,y).$$

This implies that $Q_a$ is sos, a contradiction.

A virtually identical argument shows that $q_a(\sqrt{x}, \sqrt{y}) \gg 0$ on $K_3$ for $a > 0$, but does not belong to $T_3$.

## 3 Non-compact Strips in the Plane

Before we discuss $K_1$ and $K_2$, we make a detour to $K = [-1, 1]$. There are two natural sets of generators for $K$. Let $S_1 = \{1 - x, 1 + x\}$ and $S_2 = \{1 - x^2\}$. Then clearly $K_{S_1} = K_{S_2} = K$ and $M_{S_2} = T_{S_2}$, because $|S_2| = 1$. As remarked earlier, (1) implies that $M_{S_1} = T_{S_1}$; finally, $T_{S_2} \subseteq T_{S_1}$ is immediate and

$$1 \pm x = \frac{(1 \pm x)^2}{2} + \frac{(1 - x^2)}{2} \tag{3}$$

shows the converse. Thus it does not matter whether one takes $S_1$ or $S_2$ (or $M$ or $T$) in discussing $[-1, 1]$.

What do (1) and (3) have to say in the plane? First, for $K_2$, we might take either $S_1$ or $S_2$ above as the set of generators, keeping in mind that the set of possible $\sigma$'s is taken from $\sum R_2^2$, rather than $\sum R_1^2$ as above. Then, once again $M$ and $T$ are not affected by the choice of generators and $M = T$. For $K_1$, we similarly have from (3) that $T_{\{1-x^2,y\}} = T_{\{1-x,1+x,y\}}$ and $M_{\{1-x^2,y\}} = M_{\{1-x,1+x,y\}}$. However, in this case, $T \neq M$. In fact, $y(1-x)$, which evidently is an element of $T_{\{1-x,1+x,y\}}$, does not belong to $M_{\{1-x,1+x,y\}} = M_{\{1-x^2,y\}}$.

**Theorem 2.** *Suppose $S = \{f_1(x), \ldots, f_m(x), y\}$ is such that $K_S = K_1$. Then for every $f(x) \in \mathbb{R}[x]$, we have $g(x,y) = f(x) + y(1-x) \notin M_S$. In particular, $(**)_M$ does not hold for $S$.*

*Proof.* We show that that there cannot exist an identity

$$g(x,y) = f(x) + y(1-x) = \sigma_0(x,y) + \sum_{i=1}^{m} \sigma_i(x,y) f_i(x) + \sigma_{m+1}(x,y) \cdot y, \quad (4)$$

where the $\sigma_i$'s are sos. Suppose (4) holds, and let

$$I = \{a \in [0,1) \ \mid \ \prod_i f_i(a) \neq 0\};$$

$I$ is the interval $[0,1)$ minus a finite set of points. Fix $a \in I$. Since $(a,y) \in K_1$, it follows that $f_i(a) > 0$. Consider (4) when $x = a$:

$$f(a) + y(1-a) = \sigma_0(a,y) + \sum_{i=1}^{m} \sigma_i(a,y) f_i(a) + \sigma_{m+1}(a,y) \cdot y. \quad (5)$$

Each $\sigma_i(a,y)$ is sos, and hence psd, and so as a polynomial in $y$ has leading term $c_i y^{2m_i}$, where $c_i > 0$. Let $M = \max m_i$. Then the highest power of $y$ occurring in any term on the right hand side of(5) is $y^{2M}$ or $y^{2M+1}$, with positive coefficient or coefficients, and so no cancellation occurs. In view of the left hand side, this highest power must be $y^1$. It follows that $M = 0$, so that each $\sigma_i(a,y)$ is a constant. Writing $\sigma_i(x,y) = \sum_j A_{i,j}^2(x,y)$, we see that, $\deg_y A_{i,j}(a,y) = 0$ for $a \in I$. Suppose now that $\deg_y A_{i,j}(x,y) = m_{i,j}$ and write

$$A_{i,j}(x,y) = \sum_{k=0}^{m_{i,j}} B_{i,j,k}(x) y^k,$$

We have seen that $B_{i,j,k}(a) = 0$ for $a \in I$ if $k \geq 1$. Any polynomials which vanishes on $I$ must be identically zero, hence $B_{i,j,k}(x) = 0$ for $k \geq 1$. Thus $m_{i,j} = 0$ and each $A_{i,j}(x,y)$ is, in fact, a polynomial in $x$ alone, so that $\sigma_j(x,y) = \sigma_j(x)$. Therefore, (5) becomes

$$f(x) + y(1-x) = \sigma_0(x) + \sum_{j=1}^{m} \sigma_j(x) f_j(x) + y\sigma_{m+1}(x).$$

Taking the partial derivative of both sides of this equation with respect to $y$, we see that $1 - x = \sigma_{m+1}(x)$. This contradicts the assumption that $\sigma_{m+1}$ is sos.

Let $f(x) = \epsilon$ for some $\epsilon > 0$. Then $g(x,y) = \epsilon + y(1-x)$ is positive on $K_1$, but $g \notin M$, thus $(**)_M$ fails for $K_1$. Observe, however, that if we take either of the standard generators for $K_1$, then

$$g(x,y) = \epsilon + y(1-x) = \epsilon + y \cdot \frac{(1-x)^2}{2} + \frac{y(1-x^2)}{2} \in T_S.$$

This shows that $T_S \neq M_S$ in this case. (The preceding construction works for any polynomial $f$ which is positive on $[-1,1]$.)

**Proposition 3.** *Let $K = K_2 = [-1,1] \times \mathbb{R}$ and suppose $f \in \mathbb{R}[x,y]$. The following are equivalent:*

*(i) $f \gg 0$ on $K$;*
*(ii) $f > 0$ on $K$ and $f^*(0,1,0) > 0$;*
*(iii) $f > 0$ on $K$ and the leading term of $f$ as a polynomial in $y$ is of the form $cy^d$, where $c \in \mathbb{R}$ and $d = \deg f$.*

*Proof.* It is not too hard to see that $\overline{K^*}$ consists of the intersection of the unit sphere with the set of $(u,v,w)$ satisfying $|u| \leq w$ and $w \geq 0$. Then (i) $\Rightarrow$ (ii) is clear since $(0,1,0) \in \overline{K^*}$. Suppose that (ii) holds. Let $d = \deg f$ and write $f = F_0 + \cdots + F_d$, where $F_i$ is the homogeneous part of $f$ of degree $i$, so that $f^*(x,y,z) = \sum_{j=0}^{d} z^{d-j} F_j(x,y)$. Then $f^*(0,1,0) = F_d(0,1)$, which implies $F_d(0,1) > 0$. Hence $F_d(x,y)$ must be of the form $cy^d$.

Finally, suppose that (iii) holds. We need to show that $f(u,v,w) > 0$ for $(u,v,w) \in S^2$ with $|u| < w$. If $w = 0$, then $u = 0$ and $(u,v,w) = (0,1,0)$; recall that $f(0,1,0) > 0$ by hypothesis. If $w > 0$, then $(u,v,w)$ is in $K^*$, hence $(u/w, v/w) \in K$. Since $f(u/w, v/w) > 0$, we have $f^*(u,v,w) > 0$.

Our final result is that $(*)$ holds for $K_2$. The proof uses an idea from [9]: For $g(x,y) \gg 0$ on $K_2$, fix $y = a$ and look at the one variable polynomial $g(x,a)$. This is positive on $[-1,1]$, a compact set, so we have representations of each $g(x,a)$ in $T_{1 \pm x} \subseteq \mathbb{R}[x]$. We "glue" these together to form a representation of $g(x,y)$ in $T_2$.

As in [10], for $f(x) \in \mathbb{R}[x]$ of degree $d$, we define $\tilde{f}(x)$, the *Goursat transform* of $f$, by the equation

$$\tilde{f}(x) = (1+x)^d f\left(\frac{1-x}{1+x}\right).$$

We collect some easy results from [10] about the Goursat transform:

**Lemma 1.** *If $f(x) \in \mathbb{R}[x]$ of degree $d$, then*

1. $\deg \tilde{f} \leq d$ *with equality iff* $f(-1) \neq 0$;
2. $\tilde{\tilde{f}} = 2^d f$;
3. $f > 0$ *on* $[-1,1]$ *iff* $\tilde{f} > 0$ *on* $[0,\infty)$ *and* $deg(\tilde{f}) = d$.

We also need a quantitative version of an old result, proved as [10, Theorem 6]. This is stated using the improved bound for Pólya's Theorem from [11].

**Proposition 4.** *Suppose* $f(x) = \sum_{i=0}^{d} a_i x^i \in \mathbb{R}[x]$ *and*

$$\lambda = \min\{f(x) \mid -1 \leq x \leq 1\} > 0.$$

*Let* $\tilde{f}(x) = \sum_{i=0}^{d} a_i(1-x)^i(1+x)^{d-i} = \sum_{i=0}^{d} b_i x^i$ *and let*

$$\tilde{L}(f) := \max\{|b_i| \mid i = 0, \dots d\}.$$

*Finally, let*

$$N(f) := \frac{d(d-1)}{2}\frac{\tilde{L}(f)}{\lambda}.$$

*If* $N > N(f)$*, then the coefficients of the polynomial* $(1+x)^N\tilde{f}(x)$ *are positive.*

**Theorem 3.** *Given* $N, d \in \mathbb{N}$*, there exist polynomials* $C_i \in \mathbb{R}[x_0, \dots, x_d]$*,* $0 \le i \le N+d$*, with the following property: If* $f(x) = \sum_{i=0}^{d} a_i x^i \in \mathbb{R}[x]$ *is positive on* $[-1, 1]$ *and* $N > N(f)$*, then* $C_i(a_0, \dots, a_d) > 0$ *and*

$$f(x) = \sum_{i=0}^{N+d} C_i(a_0, \dots, a_d)(1+x)^i(1-x)^{N+d-i}.$$

*Proof.* Write

$$(1+x)^N\tilde{f} = \sum_{j=0}^{N+d} b_j x^j, \tag{6}$$

where $b_j > 0$ for all $j$.

For $0 \le j \le N+d$, let $c_j(t_0, \dots, t_d)$ be the coefficient of $x^j$ in the expansion of

$$(1+x)^N \sum_{j=0}^{d} t_j(1-x)^i(1+x)^{d-i};$$

clearly each $c_j \in \mathbb{R}[t_0, \dots, t_d]$, and by construction, $b_j = c_j(a_0, \dots, a_d)$.

Now apply the Goursat transformation to both sides of (6) to obtain

$$2^{N+d}f = \sum_{j=0}^{N+d} b_j(1-x)^j(1+x)^{N+d-j}.$$

Setting $C_j = 2^{-(N+d)}c_j$, we have that $C_j(a_1, \dots, a_d) > 0$ for all $j$ and $f = \sum C_j(a_0, \dots, a_d)x^j$.

*Example 1.* For linear polynomials the proposition is easy. Suppose $f(x) = a_1x + a_0 > 0$ on $[-1, 1]$, then we can find a representation of the form specified with $N = 0$. In this case, we have

$$f(x) = C_0(a_0, a_1) \cdot (1-x) + C_1(a_0, a_1) \cdot (1-x),$$

with $C_0(t_0, t_1) = \frac{1}{2}t_0 - \frac{1}{2}t_1$ and $C_1(t_0, t_1) = \frac{1}{2}t_0 + \frac{1}{2}t_1$. Note that $f(x) > 0$ on $[-1, 1]$ implies immediately that $C_j(a_0, a_1) > 0$.

Suppose we are given $g \gg 0$ on $K_2$. For each $r \in \mathbb{R}$, define $g_r(x) \in \mathbb{R}[x]$ by $g_r(x) = g(x, r)$ and note that $g_r(x) > 0$ on $[-1, 1]$ for all $r$. Let $L_r$ denote $\tilde{L}(g_r)$ and let $\lambda_r = \min\{g_r(x) \mid -1 \leq x \leq 1\}$.

**Lemma 2.** *Suppose $g \gg 0$ on $K_2$. Then there is $u > 0$ such that*

$$\frac{\tilde{L}_r}{\lambda_r} \leq u$$

*for all $r$.*

*Proof.* This is similar to [9, Prop. 1]. Let $d = \deg_x g$ and $m = \deg_y g$, and write

$$g(x, y) = \sum_{i=0}^{m} h_i(x) y^i.$$

Since $g \gg 0$ on $K_2$, by Proposition 3, the leading term in $g$ as a polynomial in $y$, $h_m(x)$, is actually a positive real constant $c$. For each $i$, $0 \leq i \leq m-1$, there is $M_i > 0$ such that $h_i(x) < M_i$ for $x \in [-1, 1]$. Then, on $[-1, 1]$,

$$g_r(x) \geq cr^m - \sum_{j=0}^{m-1} M_j r^j > wr^m$$

for some positive constant $w$ and $|r|$ sufficiently large. In other words, for sufficiently large $|r|$, we have $\lambda_r \geq wr^m$.

Now write $g(x, y)$ as a polynomial in $x$: $g = \sum_{i=0}^{d} k_i(y) x^i$. Then $\deg k_i(y) \leq m$ for all $i$, by assumption. This means that the coefficients of $g_r(x)$ are $\mathcal{O}(|r|^m)$ as $|r| \to \infty$. The coefficients of $\tilde{g}_r(x)$ are linear combinations of the coefficients of $g_r(x)$, so the same is true for $\tilde{g}_r(x)$. From this is follows that

$$\frac{\tilde{L}_r}{\lambda_r} \leq \frac{w' r^m}{wr^m}$$

for some constant $w'$ and $|r|$ sufficiently large and the result is clear.

**Theorem 4.** $(*)$ *holds for $K_2$: If $g \gg 0$ on $K_2$, then $g \in T_2$.*

*Proof.* Let $u$ be as in the lemma and set $N = \frac{d(d-1)}{2} u$, so that we can apply Proposition 3 to each $g_r$ with this $N$.

For $i = 0, \ldots, N + d$, let $C_j \in \mathbb{R}[t_0, \ldots, t_d]$ be as in the proposition. Writing $g(x, y) = \sum_{i=0}^{d} e_i(y) x^i$, define $P_1, \ldots, P_{d+N} \in \mathbb{R}[y]$ by $P_j = C_j(e_0(y), \ldots, e_d(y))$. Then the conclusion of Theorem 3 implies that

$$g(x, y) = \sum_{j=0}^{d+N} P_j(y)(1 - x)^i (1 + x)^{N+d-i}. \tag{7}$$

For each $r \in \mathbb{R}$ and each $j$, we have $P_j(r) = C_j(e_0(r), \ldots, e_d(r))$ and then, since $\{e_0(r), \ldots, e_d(r)\}$ are the coefficients of $g_r$, it follows from the conclusion of Proposition 3 that $P_j(r) > 0$; that is $P_j > 0$ on $\mathbb{R}$ for all $j$. Thus, each $P_j(y)$ is a sum of two squares of polynomials and, plugging sos representations of the $P_j$'s into (7) yields a representation of $g$ in $T_2$.

*Example 2.* Let $g(x, y) = y^2 - xy + y + 1$, then for each $r \in \mathbb{R}$,

$$g_r(x) = -rx + (r^2 + r + 1) > 0$$

on $[-1, 1]$. By the above, we have, for each $r$, the representation

$$g_r = \frac{1}{2}(r^2 + 2r + 1)(1 - x) + \frac{1}{2}(r^2 + 1)(1 + x)$$

Then $C_0(y) = y^2+2y+1 = (y+1)^2$ and $C_1(y) = y^2+1$ yields the representation

$$g(x, y) = \frac{1}{2}(y + 1)^2(1 - x) + \frac{1}{2}(y^2 + 1)(1 + x) \in T_2$$

## References


1. M. Choi, T. Y. Lam, and B. Reznick (1995). Sums of squares of real polynomials, Proc. Sym. Pure Math. 58.2:103–126.
2. D. Hilbert (1888). Über die Darstellung definiter Formen als Summe von Formenquadraten, Math. Ann. 32: 342-350; see Ges. Abh. 2:154–161, Springer, Berlin, 1935, reprinted by Chelsea, New York 1981.
3. S. Kuhlmann and M. Marshall (2002). Positivity, sums of squares, and the multidimensional moment problem, Trans. Amer. Math Soc. 354:4285–4301.
4. J. B. Lasserre (2001). Global optimization with polynomials and the problem of moments, SIAM J. Optim. 11:796–817.
5. J.B. Lasserre (2002). SDP versus LP relaxations for polynomial programming, Math. Oper. Res. 27:347–360.
6. M. Marshall (2003). Optimization of polynomial functions, Canad. Math. Bull. 46:575–587.
7. T. Motzkin (1967). The arithmetic-geometric inequalities. In: Inequalities, O. Shisha (ed), Proc. Symp. Wright-Patterson AFB, August 19-27, 1965, 205–224. Academic Press.
8. G. Pólya and G. Szegö (1976). Problems and Theorems in Analysis II, Springer, Berlin Heidelberg New York.
9. V. Powers (2004). Positive polynomials and the moment problem for cylinders with compact cross-section, J. Pure and Appl. Alg. 188:217–226.
10. V. Powers and B. Reznick (2000). Polynomials that are positive on a interval, Trans. Amer. Math. Soc. 352:4677–4692.
11. V. Powers and B. Reznick (2001). A new bound for Pólya's Theorem with applications to polynomials positive on polyhedra, J. Pure and Applied Alg. 164:221–229.

12. M. Putinar (1993). Positive polynomials on compact semi-algebraic sets, Ind. Univ. Math. J. 969–984.
13. B. Reznick (1992). Sums of even powers of real linear forms. Mem. Amer. Math. Soc. 63, AMS, Providence, RI.
14. R. M. Robinson (1973). Some definite polynomials which are not sums of squares of real polynomials. In: Selected questions of algebra and logic (a collection dedicated to the memory of A. I. Mal'cev), Isdat. "Nauka" Sibirsk. Otdel. Novosibirsk, 264–282. Abstracts in Notices Amer. Math. Soc. 16 (1969), p. 554.
15. C. Scheiderer (2000). Sums of squares of regular functions in real algebraic varieties, Trans. Amer. Math. Soc. 352:1039–1069.
16. K. Schmüdgen (1991). The K-moment problem for compact semialgebraic sets, Math. Ann. 289:203–206.
17. G. Stengle (1996). Complexity estimates for the Schm̈udgen Positivstellensatz, J. Complexity 12:167-174.

# Stability of Interval Two–Variable Polynomials and Quasipolynomials *via* Positivity

Dragoslav D. Šiljak[1] and Dušan M. Stipanović[2]

[1] Santa Clara University, Santa Clara, CA 95053, USA; email: dsiljak@scu.edu
[2] Stanford University, Stanford, CA 94305, USA; email: dusko@stanford.edu

## 1 Introduction

Stability of two–dimensional polynomials arises in fields as diverse as 2D digital signal and image processing ([11], [7], [18]), time–delay systems [14], repetitive, or multipass, processes [24], and target tracking in radar systems. For this reason, there have been a large number of stability criteria for 2D polynomials, which have been surveyed and discussed in a number of papers ([13], [22], [9], [4], [5], [21]). In achieving the maximal efficiency of 2D stability tests, the reduction of algebraic complexity offered by the stability criteria in [26] has been useful. Apart from some minor conditions, the criteria convert stability testing of a 2D polynomial to testing of only two 1D polynomials, one for stability and the other for positivity.

Due to inherent uncertainty of the underlying models, it has been long recognized that in practical applications it is necessary to test robustness of stability to parametric variations [27]. Almost exclusively, the 2D robust stability tests have been based on the elegant Kharitonov solution of the stability problem involving 1D interval polynomials ([6], [2], [23], [16], [31]). In the context of 2D polynomials, the solution lost much of its simplicity resulting in numerically involved algorithms. This fact made the testing of 2D polynomials with interval parameters difficult, especially in the case of multiaffine and polynomic uncertainty structures.

The purpose of this paper is to present new criteria for testing stability of 2D polynomials with interval parameters, which are based on the criteria of [26] and the positivity approach to the interval parametric uncertainties advanced in [29]. An appealing feature of the new criteria is the possibility of using the efficient Bernstein minimization algorithms ([19], [8]) to carry out the numerical part of the positivity tests. Furthermore, the proposed formulation can handle the polynomic uncertainty structures having interval parameters, and can be easily extended to systems with time–delays along the lines of [14].

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 165–177, 2005.

## 2 Stability Criterion

Let us recall the stability criteria [26] for a real two–variable polynomial

$$h(s,z)=\sum_{j=0}^{n}\sum_{k=0}^{m}h_{jk}s^{j}z^{k} \tag{1}$$

where $s,z\in\mathbb{C}$ are complex variables, and for some $j,k$ the coefficients $h_{jn}$ and $h_{mk}$ are not both zero. We are interested in stating the conditions under which polynomial $h(s,z)$ satisfies the stability property

$$h(s,z)\neq 0,\quad s\in\mathbb{C}_{-}^{c}\ \cap\ z\in\mathbb{C}_{-}^{c}, \tag{2}$$

where $\mathbb{C}_{-}^{c}$ is the complement of $\mathbb{C}_{-}=\{s\in\mathbb{C}:\ \text{Re}\ s<0\}$, the open left half of the complex plane $\mathbb{C}$.

As shown by Ansell [1], property (2) is equivalent to

$$h(s,1)\neq 0,\qquad \forall s\in\mathbb{C}_{-}^{c} \tag{3a}$$

$$h(i\omega,z)\neq 0,\qquad \forall z\in\mathbb{C}_{-}^{c} \tag{3b}$$

To test (3a) we can use the standard Routh test (*e.g.*, [17]). To verify (3b) we follow Ansell's approach and consider the polynomial $c(z)=h(i\omega,z)$,

$$c(z)=\sum_{k=0}^{m}c_{k}z^{k}, \tag{4}$$

where

$$c_{k}=\sum_{j=0}^{n}h_{jk}s^{j} \tag{5}$$

and $s=i\omega$. With $c(z)$ we associate the symmetric $m\times m$ Hermite matrix $C=(c_{jk})$ with elements $c_{jk}$ defined by (*e.g.*, [17])

$$\begin{aligned}
c_{jk}&=2(-1)^{(j+k)/2}\sum_{\ell=1}^{j}(-1)^{\ell}\text{Re}\ (c_{m-\ell+1}\bar{c}_{m-j-k+\ell})\ ,\quad (j+k)\ \text{even}\\
c_{jk}&=2(-1)^{(j+k)/2}\sum_{\ell=1}^{j}(-1)^{\ell}\text{Im}\ (c_{m-\ell+1}\bar{c}_{m-j-k+\ell})\ ,\quad (j+k)\ \text{odd}
\end{aligned} \tag{6}$$

where the overbar denotes conjugacy and $j\leq k$. We recall that $C>0$ if and only if $c(z)=0$ implies $z\in\mathbb{C}_{-}$. Since $C=C(i\omega)$ is a real symmetric matrix, we define a real even polynomial

$$g(\omega^{2})=\det\ C(i\omega) \tag{7}$$

and replace $\omega^{2}$ by $\omega$ to get a polynomial $g(\omega)$.

We also define a polynomial

$$f(s) = h(s, 1) \tag{8}$$

and state the following [26]:

**Theorem 1.** *A two–variable polynomial $h(s, z)$ has the stability property (2) if and only if*

*(i)* $f(s)$ *is* $\mathbb{C}_-$*–stable.*
*(ii)* $g(\omega)$ *is* $\mathbb{R}_+$*–positive.*
*(iii)* $C(0)$ *is positive definite.*

Condition (i) means that $f(s) = 0$ implies $s \in \mathbb{C}_-$, while condition (ii) is equivalent to $g(\omega) > 0$ for all $\omega \geq 0$.

In stability analysis of recursive digital filters (*e.g.*, [11], [7]), it is of interest to establish necessary and sufficient conditions for a polynomial $h(s, z)$ to have the stability property

$$h(s, z) \neq 0, \quad \{s \in \bar{\mathbf{K}}^0\} \cap \{z \in \bar{\mathbf{K}}^0\}, \tag{9}$$

where $\mathbf{K} = \{s \in \mathbb{C} : |s| = 1\}$ is the unit circle, and $\bar{\mathbf{K}}^0 = \mathbf{K} \cup \mathbf{K}^0$ is the closure of $\mathbf{K}^0 = \{s \in \mathbb{C} : |s| < 1\}$.

By following Huang [10], one can show that (9) is equivalent to

$$h(s, 0) \neq 0, \qquad \forall s \in \bar{\mathbf{K}}^0 \tag{10a}$$

$$h(e^{i\omega}, z) \neq 0, \qquad \forall z \in \bar{\mathbf{K}}^0. \tag{10b}$$

Condition (10a) means that the new polynomial

$$f(s) = s^n h(s^{-1}, 0) \tag{11}$$

has all zeros in the open unit circle $\mathbf{K}^0$, that is, $f(s)$ is $\mathbf{K}^0$–stable. To test condition (10b), we consider

$$d(z) = z^m h(e^{i\omega}, z^{-1}) \tag{12}$$

which we write as a polynomial

$$d(z) = \sum_{k=0}^{m} d_k z^k \ , \tag{13}$$

with coefficients

$$d_k = \sum_{j=0}^{n} h_{j,m-k} s^k, \tag{14}$$

and $s = e^{i\omega}$.

With the polynomial $d(z)$ we associate the Schur–Cohn $m \times m$ matrix $D = (d_{jk})$ specified by

$$d_{jk} = \sum_{\ell=1}^{j} (d_{m-j+\ell}\bar{d}_{m-k+\ell} - \bar{d}_{j-\ell}d_{k-\ell}), \tag{15}$$

where $j \leq k$ (*e.g.*, [12]). The matrix $D(e^{i\omega})$ is a Hermitian matrix and we define

$$g(e^{i\omega}) = \det\, D(e^{i\omega}), \tag{16}$$

where $g(\cdot)$ is a self–inversive polynomial.

We state the following [26]:

**Theorem 2.** *A two–variable polynomial $h(s,z)$ has the stability property (9) if and only if*

*(i)* $f(s)$ *is* $\mathbf{K}^0$*–stable.*
*(ii)* $g(z)$ *is* $\mathbf{K}$*–positive.*
*(iii)* $D(1)$ *is positive definite.*

Positivity of $g(z)$ on $\mathbf{K}$, which is required by condition (ii), can be verified by applying the methods of [25].

Finally, we show how the mixture of the two previous stability properties can be handled using the same tools. The desired property is defined as

$$h(s,z) \neq 0, \quad \{s \in \mathbb{C}_-^c\} \cap \{z \in \bar{\mathbf{K}}^0\}\ . \tag{17}$$

By following Ansell [1], one can show that this property is equivalent to

$$h(s,0) \neq 0, \qquad \forall s \in \mathbb{C}_-^c \tag{18a}$$

$$h(i\omega,z) \neq 0, \qquad \forall z \in \bar{\mathbf{K}}^0 \tag{18b}$$

In this case, the polynomial $d(z)$ is defined as

$$d(z) = z^m h(i\omega, z^{-1})\ , \tag{19}$$

which is used to obtain the polynomial $g(\cdot)$ *via* equations (13)–(16). From (18a), we get the polynomial

$$f(s) = h(s,0), \tag{20}$$

then define $\mathbf{I} = \{z \in \mathbb{C} : \text{Re } z = 0\}$ and arrive at [26]:

**Theorem 3.** *A two–variable polynomial $h(s,z)$ has the stability property (17) if and only if*

*(i)* $f(s)$ *is* $\mathbb{C}_-$*–stable.*
*(ii)* $g(z)$ *is* $\mathbf{I}$*–positive.*
*(iii)* $D(0)$ *is positive definite.*

We note that $\mathbf{I}$–positivity of $g(z)$ can be reformulated as $\mathbb{R}_+$–positivity (see [28]).

## 3 Uncertain Polynomials

We are now interested in studying stability properties of uncertain two–variable polynomials with polynomic uncertainty structures. A polynomial $h(s, z; p)$ is given as

$$h(s, z; p) = \sum_{j=0}^{n} \sum_{k=0}^{m} h_{jk}(p) s^j z^k, \tag{21}$$

where $h_{jk}(p)$ are polynomials themselves in uncertain parameter vector $p \in \mathbb{R}^r$. We assume that $p$ resides in a box

$$\mathbf{P} = \{p \in \mathbb{R}^r : \quad p_k \in [\underline{p}_k, \bar{p}_k], k \in \mathbf{r}\} \ . \tag{22}$$

We want to obtain the robust versions of stability properties defined in the preceding section. In the case of (2), for example, we are interested in testing the robust property

$$h(s, z; p) \neq 0, \quad \{s \in \mathbb{C}_-^c\} \cap \{z \in \mathbb{C}_-^c\} \cap \{p \in \mathbf{P}\}. \tag{23}$$

To accommodate the uncertainty in $h(s, z; p)$ we define the polynomial families $\mathcal{F} = \{f(\cdot, p) \ : \ p \in \mathbf{P}\}$, $G = \{g(\cdot, p) : p \in \mathbf{P}\}$ and state a straightforward modification of Theorem 1.

**Theorem 4.** *An uncertain two–variable polynomial* $h(s, z; p)$ *has the robust stability property (23) if and only if*

*(i)* $\mathcal{F}$ *is* $\mathbb{C}_-$*–stable.*
*(ii)* $\mathcal{G}$ *is* $\mathbb{R}_+$*–positive.*
*(iii)* $C(0, p)$ *is positive definite for all* $p \in \mathbf{P}$*.*

Robust versions of the remaining two stability properties of the preceding section can be tested by Theorem 4 *via* bilinear transformation in pretty much the same way $D$–stability was tested in [29]. We also note the structural similarity of Theorem 4 with theorems on robust SPR properties [30], which motivates the work presented next.

Condition (i) in Theorem 4 obviously means that all zeros of $f(s,p)$ lie in $\mathbb{C}_-$ for all $p \in \mathbf{P}$. To establish this type of robust stability *via* polynomial positivity, we define the magnitude function

$$\hat{f}(s,p) = f(s,p)\overline{f(s,p)} = \sum_{k=0}^{n}\sum_{j=0}^{n} a_k(p)\bar{a}_j(p)s^l\bar{s}^j \tag{24}$$

where overbar denotes conjugation. We note immediately that the magnitude function $\hat{f}(s,p) = |f(s,p)|^2$ is nonnegative for all $s \in \mathbb{C}$. This obvious fact is essential in the following development.

Let us form a family $\hat{\mathcal{F}} = \{\hat{f}(\cdot,p) : \ p \in \mathbf{P}\}$ and use the result of [29] to conclude that the family $\mathcal{F}$ is $\mathbb{C}_-$–stable if and only if the corresponding family $\hat{\mathcal{F}}$ is $\mathbf{I}$–positive, and $f(s,p')$ is $\mathbb{C}_-$–stable for some $p' \in \mathbf{P}$. Furthermore, from (7) it follows that positivity of $\det C(0;p)$ is included in testing condition (ii) of Theorem 4. This means that to test condition (iii) of Theorem 4, it suffices to verify that $C(0;p'')$ is positive definite for some $p'' \in \mathbf{P}$. We finally arrive at

**Theorem 5.** *An uncertain two–variable polynomial $h(s,z;p)$ has the robust stability property (23) if and only if*

*(i)* $\hat{\mathcal{F}}$ *is $\mathbb{R}_+$–positive and $f(s;p')$ is $\mathbb{C}_-$–stable for some $p' \in \mathbf{P}$.*
*(ii)* $\mathcal{G}$ *is $\mathbb{R}_+$–positive.*
*(iii)* $C(0;p'')$ *is positive definite for some $p'' \in \mathbf{P}$.*

*Example 1.* To illustrate the application of Theorem 5, let us use the two–variable polynomial from [31],

$$h(s,z;p) = h_{11}(p)sz + h_{10}(p)s + h_{01}(p)z + h_{00}(p) \ , \tag{25}$$

where

$$\begin{aligned} h_{11}(p) &= 0.9 - 0.1p_1 - 0.3p_2 \\ h_{10}(p) &= 0.8 - 0.5p_1 + 0.3p_2 \\ h_{01}(p) &= 1 + 0.2p_1 + 0.3p_2 \\ h_{00}(p) &= 1.6 + 0.5p_1 - 0.7p_2 \end{aligned} \tag{26}$$

and

$$\mathbf{P} = \{p \in \mathbb{R}^2; \ \ p_1 \in [-0.3, 0.4], \ \ p_2 \in [0.1, 0.5]\} \ . \tag{27}$$

To test condition (i) we compute the polynomial

$$f(s;p) = (1.7 - 0.6p_1)s + 2.6 + 0.7p_1 - 0.4p_2 \tag{28}$$

and note that, in this simple case, we do not need to compute the corresponding magnitude function $\hat{f}(\omega;p)$. Robust $\mathbb{C}_-$–stability of $f(s;p)$ follows directly from positivity of its coefficients. Indeed,

$$\begin{aligned} &1.7 - 0.6p_1 \geq 1.46 \\ &2.6 + 0.7p_1 - 0.4p_2 \geq 2.19 \end{aligned} \tag{29}$$

for all $p \in \mathbf{P}$.

Since the matrix $C(i\omega; p)$ is a scalar, condition (iii) is included in condition (ii) which is satisfied because

$$\begin{aligned} g(\omega; p) &= (1 + 0.2p_1 + 0.3p_2)(1.6 + 0.5p_1 - 0.7p_2) \\ &\quad + (0.9 - 0.1p_1 - 0.3p_2)(0.8 - 0.5p_1 + 0.3p_2)\omega \\ &\geq 0.4473\omega + 1.0670 \end{aligned} \tag{30}$$

is obviously $\mathbb{R}_+$–positive.

Our analysis is elementary when compared to the stability testing procedure of Xiao [31], which involves extensive computation required by the Edge Theorem.

Let us consider more complex examples which will require the use of Bernstein's algorithm.

*Example 2.* A two–variable polynomial is given as

$$\begin{aligned} h(s, z; p) =& s^2z^2 + h_{21}(p)s^2z + h_{12}(p)sz^2 + h_{20}(p)s^2 \\ &+ h_{02}(p)z^2 + h_{11}(p)sz + h_{10}(p)s + h_{01}(p)z + h_{00}(p) \end{aligned} \tag{31}$$

where

$$\begin{aligned} h_{21}(p) &= 3 - p_1 \\ h_{20}(p) &= p_1p_2 \\ h_{11}(p) &= 3p_1p_2 - p_1^2p_2 \\ h_{01}(p) &= 6 - 5p_1 + 3p_2 - p_1p_2 + p_1^2 \\ h_{12}(p) &= p_1p_2 \\ h_{02}(p) &= 2 - p_1 + p_2 \\ h_{10}(p) &= p_1^2p_2^2 \\ h_{00}(p) &= 2p_1p_2 - p_1^2p_2 + p_1p_2^2 \end{aligned} \tag{32}$$

and

$$\mathbf{P} = \{p \in \mathbb{R}^2 : \quad p_1 \in [1, 2], \quad p_2 \in [1, 2]\}. \tag{33}$$

The polynomial $f(s; p)$ is computed as

$$\begin{aligned} f(s; p) = (4 - p_1 + p_1p_2)s^2 &+ (4p_1p_2 - p_1^2p_2 + p_1^2p_2^2)s \\ &+ 8 - 6p_1 + 4p_2 + p_1p_2 + p_1^2 - p_1^2p_2 + p_1p_2^2 \end{aligned} \tag{34}$$

The corresponding minimizing polynomial

$$\underline{f}(s) = 4s^2 + 4s + 4 \tag{35}$$

is obtained by minimizing each coefficient using Bernstein's algorithm. Obviously, $\mathcal{F}$ is $\mathbb{C}_-$–stable since $\underline{f}(s)$ has positive coefficients.

Next, we compute

$$c(z;p) = c_2(p)z^2 + c_1(p)z + c_0(p) \ , \tag{36}$$

where

$$\begin{aligned} c_2(p) &= 2 - p_1 + p_2 - \omega^2 + ip_1p_2\omega \\ c_1(p) &= 6 - 5p_1 + 3p_2 - p_1p_2 + p_1^2 - 3\omega^2 + p_1\omega^2 + i(3p_1p_2 - p_1^2p_2)\omega \\ c_0(p) &= 2p_1p_2 - p_1^2p_2 - p_1p_2\omega^2 + ip_1^2p_2^2\omega \ . \end{aligned} \tag{37}$$

In this case, the $2 \times 2$ matrix $C(i\omega;p)$ turns out to be a diagonal matrix

$$C(i\omega;p) = \text{diag } \{c_{11}(i\omega;p), \quad c_{22}(i\omega;p)\} \tag{38}$$

and conditions (ii) and (iii) of Theorem 5 reduce to positivity of the coefficients

$$\begin{aligned} c_{11}(i\omega;p) &= \tilde{c}_{11}(\omega^2;p) \\ &= (3 - p_1)\omega^4 + (-12 + 10p_1 - 6p_2 + 2p_1p_2 - 2p_1^2 \\ &\quad + 3p_1^2p_2^2 - p_1^3p_2^2)\omega^2 + 12 - 16p_1 + 12p_2 - 10p_1p_2 \\ &\quad + 7p_1^2 + 3p_2^2 - p_1p_2^2 + 2p_1^2p_2 - p_1^3 \\ c_{22}(i\omega;p) &= \tilde{c}_{22}(\omega^2;p) \\ &= (3p_1p_2 - p_1^2p_2)\omega^4 + (-12p_1p_2 + 10p_1^2p_2 - 6p_1p_2^2 \\ &\quad + 2p_1^2p_2^2 - 2p_1^3p_2 + 3p_1^3p_2^3 - p_1^4p_2^3)\omega^2 \\ &\quad + 12p_1p_2 - 16p_1^2p_2 + 12p_1p_2^2 - 10p_1^2p_2^2 + 7p_1^3p_2 + 3p_1p_2^3 \\ &\quad - p_1^2p_2^3 + 2p_1^3p_2^2 - p_1^4p_2 \ . \end{aligned} \tag{39}$$

By using Bernstein's minimization algorithm we compute the minorizing polynomials [29] and establish positivity of the two polynomials $c_{11}(i\omega;p)$ and $c_{22}(i\omega;p)$ by obtaining the minima

$$\begin{aligned} &\min_{\substack{p\in\mathbf{P} \\ \omega\in\mathbb{R}_+}} \tilde{c}_{11}(\omega;p) = 1.3724 \text{ at } \ p_1 = 2, \ \ p_2 = 1, \ \omega = 0.1715 \\ &\min_{\substack{p\in\mathbf{P} \\ \omega\in\mathbb{R}_+}} \tilde{c}_{22}(\omega;p) = 3.1185 \text{ at } \ p_1 = 2, \ \ p_2 = 1, \ \omega = 0.2487 \ . \end{aligned} \tag{40}$$

Positivity of the minima implies robust stability property (25) for the polynomial $h(s,z;p)$ of (31).

## 4 Time–Delay Systems

Our objective in this section is to show how the tools presented in this chapter can be applied to test robust stability of linear systems of the retarded type described by a differential–difference equation

$$x^{(n)}(t) + \sum_{j=0}^{n-1} \sum_{k=0}^{m} h_{jk}(p) x^{(j)}(t - k\tau) = 0 \tag{41}$$

where $\tau > 0$ is a fixed delay. The coefficients $h_{jk}(p)$ are polynomials in the uncertain parameter vector $p \in \mathbb{R}^{\ell}$ which belongs to a box $\mathbf{P}$.

It is well known [3] that for a fixed parameter $p$, a system (41) is stable if and only if the corresponding quasipolynomial satisfies the following property:

$$h(s, e^{-\tau s}; p) = s^n + \sum_{j=0}^{n-1} \sum_{k=0}^{m} h_{jk}(p) s^j e^{-k\tau s} \neq 0 \ , \ \text{Re } s \geq 0 \ . \tag{42}$$

The system is robustly stable if (42) holds for all $p \in \mathbf{P}$.

The following theorem is a straightforward robustification of a theorem by Kamen ([14], [15]):

**Theorem 6.** *System (41) is robustly stable independent of delay if*

$$h(s, z; p) \neq 0, \quad \{s \in \mathbb{C}_{-}^{c}\} \cap \{z \in \mathbf{K}\} \cap \{p \in \mathbf{P}\} \ . \tag{43}$$

This condition is also necessary if

$$h(0, z; p) \neq 0, \quad \{z \in \mathbf{K}\} \cap \{p \in \mathbf{P}\} \ . \tag{44}$$

To test condition (43) we first use the bilinear transformation

$$z = \begin{cases} \dfrac{1 + i\omega}{1 - i\omega} \ , & \omega \in \mathbb{R} \text{ when } z \in \mathbf{K} \backslash \{-1\} \\ -1 \ , & z = -1 \end{cases} \tag{45}$$

to define the polynomials

$$\begin{aligned} \tilde{h}(s, i\omega; p) &= (1 - i\omega)^m h(s, \frac{1 + i\omega}{1 - i\omega}; p) \\ f(s; p) &= h(s, -1; p) \ . \end{aligned} \tag{46}$$

Then, by following Ansell's approach [1], we consider the polynomial $c(s; p) = \tilde{h}(s, i\omega; p)$,

$$c(s; p) = \sum_{j=0}^{n} c_j(p) s^j \tag{47}$$

where

$$c_j(p) = \sum_{k=0}^{n} \tilde{h}_{jk}(p)(i\omega)^k \ . \tag{48}$$

With $c(s;p)$ we associate the symmetric $n \times n$ Hermite matrix $C = (c_{jk})$ having elements $c_{jk}$ defined in (6), and obtain the polynomial

$$g(\omega^2;p) = \det\, C(i\omega;p) \ . \tag{49}$$

Finally, with polynomials $f(s;p)$ and $g(\omega^2;p)$ at hand, we can imitate Theorem 5 to state the following:

**Theorem 7.** *System (41) is robustly stable independent of delay if*

*(i)* $\hat{\mathcal{F}}$ *is* $\mathbb{R}_+$*–positive and* $f(s;p')$ *is* $\mathbb{C}_-$*–stable for some* $p' \in \mathbf{P}$.
*(ii)* $\mathcal{G}$ *is* $\mathbb{R}_+$*–positive.*
*(iii)* $C(0;p'')$ *is positive definite for some* $p'' \in \mathbf{P}$.

It is obvious that condition (44) of Theorem 6, which is included in (45), can be tested *via* positivity as well.

To illustrate the application of Theorem 7 let us use the following:

*Example 3.* A time–delay system (41) is given as

$$x^{(2)}(t) + p_2 x^{(1)}(t-\tau) + p_1 x(t-\tau) + x^{(1)}(t) + (1 + p_1 p_2^2)x(t) = 0 \tag{50}$$

with the uncertainty box

$$\mathbf{P} = \{p \in \mathbb{R}^2 : \quad p_1 \in [-0.5, 0.5], \quad p_2 \in [-0.5, 0.5]\} \ . \tag{51}$$

From (50), we compute the associated quasipolynomial

$$h(s,z;p) = s^2 + (p_2 s + p_1)z + s + 1 + p_1 p_2^2 \ , \tag{52}$$

and test first the necessity of condition (43) by checking condition (44). Since

$$h(0,z;p) = p_1 z + 1 + p_1 p_2^2 \tag{53}$$

and $1 + p_1 p_2^2 > |p_1|$, we conclude that (44) is satisfied. This implies that condition (43) is necessary and sufficient for robust stability of system (50), and we proceed to compute the polynomial

$$f(s;p) = s^2 + (1 - p_2)s + 1 - p_1 + p_1 p_2^2 \ . \tag{54}$$

To test robust stability of this polynomial we do not need to construct the family $\hat{\mathcal{F}}$. It suffices to check positivity of each coefficient, which we do by using the Bernstein algorithm. The resulting minorizing polynomial

$$\underline{f}(s) = s^2 + 0.5s + 0.5 \tag{55}$$

implies robust stability of $f(s;p)$, that is, condition (i) of Theorem 7 is satisfied.

For testing condition (ii) we need the polynomial

$$\begin{aligned}\tilde{h}(s,i\omega;p) = (1-i\omega)s^2 + [1+p_2+(-1+p_2)i\omega]s \\ + 1 + p_1 + p_1p_2^2 + (-1+p_1-p_1p_2^2)i\omega \ .\end{aligned} \tag{56}$$

Using equations (47)–(49), we compute

$$\begin{aligned}g(\omega;p) = 4(1-p_1-2p_2+2p_2^2+2p_1p_2-2p_1p_2^3+p_1p_2^4)\omega^2 \\ + 8(1-2p_1^2-p_2^2+p_1p_2^2-p_1p_2^4)\omega \\ + 4(1+p_1+2p_2+2p_1p_2+p_2^2+2p_1p_2^2+2p_1p_2^3+p_1p_2^4.\end{aligned} \tag{57}$$

By applying the Bernstein algorithm to each coefficient of $g(\omega;p)$, we obtain the minorizing polynomial

$$\underline{g}(\omega) = 0.625\omega^2 + 1.25\omega + 0.375 \ , \tag{58}$$

which is clearly $\mathbb{R}_+$–positive, and (ii) of Theorem 7 is satisfied.

Finally, the matrix of condition (iii) is computed as $C(0,0) = 2I_2$, where $I_2$ is the identity matrix of dimension 2, and robust stability independent of delay of system (50) is established with respect to the uncertainty box $\mathbf{P}$ in (51).

## 5 Conclusions

We have shown how stability of 2D polynomials and quasipolynomials with interval parameters can be tested *via* polynomial positivity. To test stability of polynomials with multiaffine and polynomic uncertainty structures, positivity of only two interval polynomials is required. A remarkable efficiency of the proposed stability criteria is due to their suitability for applications of Bernstein's expansion algorithms.

**Acknowledgement.**
The research reported herein has been supported by the National Science Foundation under the grant ECS-0099469.

## References


1. Ansell H. G. (1964). On certain two–variable generalizations of circuit theory with applications to networks of transmission lines and lumped reactances. IEEE Transactions on Circuit Theory, 11:214–223.
2. Basu S. (1990). On boundary implications of stability and positivity properties of multidimensional systems. Proceedings of IEEE, 78:614–626.
3. Bellman R. E., and Cooke K. L. (1963). Differential-Difference Equations, Academic Press, New York.
4. Bistritz Y. (1999). Stability testing of two–dimensional discrete linear system polynomials by a two–dimensional tabular form. IEEE Transactions on Circuits and Systems, Part 1, 46:666-667.
5. Bistritz Y. (2000). Immitance–type tabular stability for 2–D LSI systems based on a zero location test for 1–D complex polynomials. Circuits, Systems and Signal Processing, 19:245–265.
6. Bose N. K., and Zeheb E. (1986). Kharitonov's theorem and stability test of multidimensional digital filters. IEE Proceedings, 133:187–190.
7. Dudgeon D., and Mersereau R. (1984). Multidimensional Digital Signal Processing. Prentice–Hall, Englewood Cliffs, NJ.
8. Garloff J. (1993). Convergent bounds for the range of multivariable polynomials. Lecture Notes in Computer Science, 212:37–56, Springer, Berlin.
9. Hu X. (1994). Techniques in the stability testing of discrete time systems. In C. T. Leondes (Ed.). Control and Dynamic Systems. 66:153–216. Academic Press, San Diego, CA.
10. Huang T. S. (1972). Stability of two-dimensional recursive filters. IEEE Transactions on Audio and Electroacustics, 20:158–163.
11. Huang T. S. (1981). Two–Dimensional Digital Signal Processing I. Springer, Berlin.
12. Jury E. I. (1982). Inners and Stability of Dynamic Systems. (2nd ed.). Kreiger, Melbourne, FL.
13. Jury E. I. (1986). Stability of multidimensional systems and related problems. In S. G. Tzafestas (ed.). Multidimensional systems, Techniques and Applications, 89–159, Marcel Dekker, NY.
14. Kamen E. W. (1982). Linear systems with commensurate time delays: Stability and stabilization independent of delay. IEEE Transactions on Automatic Control, 27:367–375.
15. Kamen E. W. (1983). Correction to "Linear systems with comensurate time delays: Stability and stabilization independent of delay". IEEE Transactions on Automatic Control, 28:248–249.
16. Kharitonov V. L., Torres Munoz J. A., and Ramirez-Sosa M. I. (1997). Stability and robust stability of multivariable polynomials. Proc. IEEE Conf. on Decision and Control, 3254–3259.
17. Lehnigh S. H. (1966). Stability Theorems for Linear Motions. Prentice–Hall, Englewood Cliffs, NJ.
18. Lim J. S. (1990). Two–Dimensional Signal and Image Processing. Prentice–Hall, Englewood Cliffs, NJ.
19. Malan S., Milanese M., Taragna M., and Garloff J. (1992). $B^3$ algorithm for robust performance analysis in presence of mixed parametric and dynamic perturbations, Proc. IEEE Conf. on Decision and Control, 128–133.

20. Marden M. (1966). Geometry of Polynomials. AMS, Providence, RI.
21. Mastorakis N. E. (2000). New necessary stability conditions for 2–D systems. IEEE Transactions on Circuits and Systems, Part 1, 47:1103–1105.
22. Premaratne K. (1993). Stability determination of two–dimensional discrete–time systems. Multidimensional Systems and Signal Processing, 4:331–354.
23. Rajan P. K., and Reddy H. C. (1991). A study of two–variable very strict Hurwitz polynomials with interval coefficients. Proc. IEEE Intl. Symposium on Circuits and Systems, 1093–1096.
24. Rogers E., and Owens D. H. (1993). Stability tests and performance bounds for a class of 2D linear systems. Multidimensional Systems and Signal Processing, 4:355–391.
25. Šiljak D. D. (1973). Algebraic criteria for positive realness relative to the unit circle. Journal of the Franklin Institute, 296:115–122.
26. Šiljak D. D. (1975). Stability criteria for two–variable polynomials. IEEE Transactions on Circuits and Systems, CAS–22:185–189.
27. Šiljak D. D. (1989). Parameter space methods for robust control design: A guided tour. IEEE Transactions on Automatic Control, 34:674–688.
28. Šiljak D. D., and Šiljak M. D. (1998). Nonnegativity of uncertain polynomials. Mathematical Problems in Engineering, 4:135–163.
29. Šiljak D. D., and Stipanović D. M. (1999). Robust $D$–stability via positivity. Automatica, 35:1477–1484.
30. Stipanović D. M., and Šiljak D. D. (2001). SPR criteria for uncertain rational matrices via polynomial positivity and Bernstein's expansions. IEEE Transactions of Circuits and Systems, Part I, 48:1366–1369.
31. Xiao Y., Unbehauen R., Du X. (1999). Robust Hurwitz stability conditions of polytopes of bivariate polynomials. Proc. IEEE Conf. on Decision and Control, 5030–5035.

# Exploiting Algebraic Structure in Sum of Squares Programs

Pablo A. Parrilo

Automatic Control Laboratory
Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland
parrilo@control.ee.ethz.ch

We present an overview of several different techniques available for exploiting structure in the formulation of semidefinite programs based on the sum of squares decomposition of multivariate polynomials. We identify different kinds of algebraic properties of polynomial systems that can be successfully exploited for numerical efficiency. Our results here apply to three main cases: sparse polynomials, the ideal structure present in systems with explicit equality constraints, and structural symmetries, as well as combinations thereof. The techniques notably improve the size and numerical conditioning of the resulting SDPs, and are illustrated using several control-oriented applications.

## 1 Introduction

From an abstract computational viewpoint, the branch of mathematics best suited to deal with a large fraction of the robustness analysis and synthesis problems of control theory is *real algebraic geometry*. This discipline takes as one of its main objects of study the solution set of polynomial equations and inequalities over the reals, the so-called *semialgebraic sets*.

While a significant part of the computational algebra literature has dealt extensively with the development of algorithmic tools for the manipulation of semialgebraic sets (for instance, see [2, 15]), their use within the control community has been sporadic (a few examples being [1, 9, 12]). Undoubtedly, good motives for this are the demanding computational requirements of purely algebraic methods. These, in turn, seem to be a necessary consequence of the inherent hardness of the underlying problems, as well as a side effect of the strict requirements on an exact computational representation of the solution.

For good practical reasons such as the ones mentioned, the main tools of the control community have been slanted much more towards purely numerical computation. The resounding success of convex optimization based approaches to many control problems [4, 5] is but just one illustration of this phenomenon.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 181–194, 2005.

It is our viewpoint that there is much to be gained from bringing these two nearly separate mindsets together. In this direction, in [17] we have recently introduced a computational framework based on sum of squares (SOS) decompositions of multivariate polynomials, that enables the use of semidefinite programming (SDP) towards a full characterization of the feasibility and solutions of polynomial equations and inequalities. This combination of concepts from convex optimization and computational algebra has not only proven to be extremely powerful in several control problems such as computation of Lyapunov functions for nonlinear systems [17, Chapter 7], but also through new applications in very diverse fields, such as combinatorial optimization and quantum mechanics.

Despite the theoretical elegance of the sum of squares techniques, it is clearly the case that for efficient practical performance in medium- or large-scale problems, it becomes necessary to take a deeper look at the inherent structure of the problem at hand. This is the motivation of the present work, where we identify and isolate several abstract algebraic properties of a polynomial system that are amenable to be exploited through the SOS/SDP machinery. While other algebraic techniques can (and do) exploit certain kinds of structural features, our results show that the flexibility of convex optimization allows in this case for a much higher degree of customization. Throughout the paper, the main ideas of each approach are outlined in a concise way for obvious space reasons, and are illustrated with simple, but representative examples.

A deeper issue related to these ideas, is the extent to which algebraic properties of the input polynomial system are inherited by the corresponding infeasibility certificates. The results in [10], for instance, show that symmetries in the input data induce similar symmetries in the solution of the SDPs, and this property can be successfully exploited. We will see a concrete instance of this in the examples in section 7.

A description of the paper follows: in Section 2 we give an overview of the basic SOS/SDP methods in the simplest case, that of dense polynomials. In Section 3 we present the simplifications that are possible when the polynomials at hand are sparse, in the sense of having "small" Newton polytopes. In the following section we illustrate the reduction in the number of variables and SDP size that are possible when equality constraints are present, by working on the quotient ring. In Section 5 we discuss the case of symmetries, to finally conclude with a few comments on the possibilities of combining the different approaches and a control-oriented example.

## 2 The Basic SOS/SDP Techniques

We present next a brief description of the main ideas behind the use of semidefinite programming in the computation of sum of squares decompositions, as well as its use in the certification of properties of basic semialgebraic sets. We

refer the reader to [17, 19] (and the references therein) for a more thorough introduction of the SOS/SDP machinery and applications, as well as to the related work in [16, 13, 6].

An obvious sufficient condition for non-negativity of a polynomial is the existence of a representation as a sum of squares of other polynomials. The connections between sums of squares and non-negativity have been extensively studied since the end of the 19th century, when Hilbert showed that in the general case these two properties are not equivalent. The work of Choi, Lam, and Reznick [7] presented a full analysis of the geometric structure of sums of squares decompositions, and the important "Gram matrix" method was formally introduced, already implicitly present in several of the authors' earlier works. A parallel development focusing on the convex optimization side appears in the early work of Shor [24]. Recently, new results and the use of efficient techniques based on semidefinite programming have given new impulse to this exciting research area.

Consider a multivariate polynomial $F(\mathbf{x})$ for which we want to decide whether a sum of squares decomposition exists. As we will see, this question is reducible to semidefinite programming, because of the following result:

**Theorem 1.** *A multivariate polynomial $F(\mathbf{x})$ is a sum of squares if and only if*

$$F(\mathbf{x}) = \mathbf{u}^T Q \mathbf{u}, \tag{1}$$

*where $\mathbf{u}$ is a vector whose entries are monomials in the $x_i$ variables, and $Q$ is a symmetric positive semidefinite matrix.*

An efficient choice of the specific set of monomials $\mathbf{u}$ will depend on both the sparsity structure and symmetry properties of $F$. For the simplest case of a generic dense polynomial of total degree $2d$, the variables $\mathbf{u}$ can always be taken to be all the monomials (in the variables $x_i$) of degree less than or equal to $d$. Since in general the variables $\mathbf{u}$ will not be algebraically independent, the matrix $Q$ in the representation (1) *is not unique*. In fact, there is an affine subspace of matrices $Q$ that satisfy the equality, as can be easily seen by expanding the right-hand side and equating term by term. When finding a SOS representation, we need to find a positive semidefinite matrix in this affine subspace. Therefore, the problem of checking if a polynomial can be decomposed as a sum of squares is *equivalent* to verifying whether a certain affine matrix subspace intersects the cone of positive definite matrices, and hence an SDP feasibility problem.

*Example 1.* Consider the quartic form in two variables described below, and define $\mathbf{u} = [x^2, y^2, xy]^T$.

$$\begin{aligned} F(x,y) &= 2x^4 + 2x^3y - x^2y^2 + 5y^4 \\ &= \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix} \\ &= q_{11}x^4 + q_{22}y^4 + (q_{33} + 2q_{12})x^2y^2 \\ &\quad + 2q_{13}x^3y + 2q_{23}xy^3 \end{aligned}$$

Therefore, for the left- and right-hand sides to be identical, the following linear equalities should hold:

$$q_{11} = 2, \quad q_{22} = 5, \quad q_{33} + 2q_{12} = -1, \quad 2q_{13} = 2, \quad 2q_{23} = 0. \tag{2}$$

A positive semidefinite $Q$ that satisfies the linear equalities can then be found using semidefinite programming. A particular solution is given by:

$$Q = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} = L^T L, \quad L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix},$$

and therefore we have the sum of squares decomposition:

$$F(x,y) = \frac{1}{2}(2x^2 - 3y^2 + xy)^2 + \frac{1}{2}(y^2 + 3xy)^2.$$

□

While polynomial nonnegativity is an important property *per se*, the strength of the SOS/SDP ideas is that the same machinery can be extended to the much larger class of semialgebraic problems, i.e., those that can be describe with a finite number of polynomial equalities and inequalities.

The key result here is a central theorem in real algebraic geometry, usually called *Positivstellensatz* and due to Stengle [3], that gives a full characterization of the infeasibility of systems of polynomial equations and inequalities over the reals. The theorem states that if a system of polynomial inequalities is infeasible, then there exists a particular algebraic identity that proves (in an obvious way) that this is the case.

*Example 2.* Here is a very simple example (taken from [21]) to illustrate the idea of Positivstellensatz certificates. Consider the system:

$$f := x - y^2 + 3 \geq 0, \qquad h := y + x^2 + 2 = 0. \tag{3}$$

We want to prove that the system is inconsistent, i.e., there are no $(x,y) \in \mathbb{R}^2$ that satisfy (3). By the Positivstellensatz, the system $\{f \geq 0, h = 0\}$ has no solution, if and only if there exist polynomials $s_1, s_2, t_1 \in \mathbb{R}[x,y]$ that satisfy the following:

$$s_1 + s_2 \cdot f + t_1 \cdot h \equiv -1\,, \quad \text{where } s_1 \text{ and } s_2 \text{ are SOS.} \tag{4}$$

Sufficiency of this condition should be obvious: evaluating the expression above at any candidate feasible point of (3) yields a contradiction, since the left-hand side would be nonnegative, while the left-hand side is negative. Possible values of the polynomials certifying the inconsistency of the system are:

$$s_1 = \tfrac{1}{3} + 2\left(y + \tfrac{3}{2}\right)^2 + 6\left(x - \tfrac{1}{6}\right)^2, \quad s_2 = 2, \quad t_1 = -6.$$

It can be easily verified that these polynomials satisfy the identity (4), and therefore prove the inconsistency of the system $\{f \geq 0,\, h = 0\}$.

Positivstellensatz refutations are purely algebraic certificates of emptiness, for which the verification process is immediate (compare this with SOS decompositions as "easy" certificates of nonnegativity). The reason why this relevant in the context of this paper, of course, is because given a degree bound, we can compute these refutations using semidefinite programming [17, 19]. To see this in our example, notice that condition (4) is *affine* in the polynomial unknowns $s_i$, and therefore can be naturally combined with the Gram matrix procedure described earlier.

Clearly, once we have a way to address emptiness of semialgebraic sets in a computationally attractive way, there are many problems that fall into this domain through more or less obvious reformulations. Particularly important are standard optimization problems (by considering the emptiness of sublevel sets), set intersections, set propagation under nonlinear mappings, etc.

## 3 Sparsity

In the general dense case, multivariate polynomials can have a very large number of coefficients. It is well-known, and easy to verify, that a dense polynomial in $n$ variables of degree $d$ has $\binom{n+d}{d}$ coefficients. Even for relatively small values of $n, d$ this can be a very large number. Nevertheless, most of the higher degree polynomials that appear in practice usually have a lot of additional structure. Just like most large-scale matrices of practical interest are *sparse*, a similar notion is appropriate in the polynomial case.

For standard matrices the notion of sparsity commonly used is relatively straightforward, and relates only to the number of nonzero coefficients. In computational algebra, however, there exists a much more refined notion of sparsity. This notion is linked to the so-called *Newton polytope* of a polynomial, defined as the convex hull of the set of exponents, considered as vectors in $\mathbb{R}^n$.

*Example 3.* Consider the polynomial $p(x, y) = 1 - x^2 + xy + 4y^4$. Its Newton polytope $\mathrm{New}(p)$ is the triangle in $\mathbb{R}^2$ with vertices $\{(0, 0), (2, 0), (0, 4)\}$.

Newton polytopes are an essential tool when considering polynomial arithmetic because of the following identity:

$$\mathrm{New}(g \cdot h) = \mathrm{New}(g) + \mathrm{New}(h),$$

where + is the Minkowski addition of polytopes.

Sparsity (in this algebraic sense) allows a notable reduction in the computational cost of checking sum of squares conditions. The reason is the following theorem due to Reznick:

**Theorem 2 ([23], Theorem 1).** *If* $F = \sum g_i^2$, *then* $\mathrm{New}(g_i) \subseteq \frac{1}{2}\mathrm{New}(F)$.

*Example 4.* Consider the following polynomial, taken from the SOSTOOLS [22] manual:

$$F = (w^4 + 1)(x^4 + 1)(y^4 + 1)(z^4 + 1) + 2w + 3x + 4y + 5z.$$

The polynomial $F$ has degree $2d = 16$, and four independent variables ($n = 4$). A naive approach, along the lines described earlier, would require a matrix of size $\binom{n+d}{d} = 495$. However, the Newton polytope of $F$ is easily seen to be the four-dimensional hypercube with vertices at $(0, 0, 0, 0)$ and $(4, 4, 4, 4)$. Therefore, the polynomials $g_i$ in the SOS decomposition of $F$ will have at most $3^4 = 81$ distinct monomials, and as a consequence the full SOS decomposition can be computed by solving a much smaller SDP.

## 4 Equality Constraints

The Positivstellensatz refutations approach, discussed earlier, attempts to find certificates of infeasibility of systems of polynomial equations and inequalities. When explicit equality constraints are present in the problem, then some notable simplifications in the formulation of the SDPs are possible. We explore these ideas in this section.

Let $I$ be the ideal defined by the equality constraints, and define the quotient ring $\mathbb{R}[x]/I$ as the set of equivalence classes for congruence modulo $I$. Then, provided computations can be effectively done in this quotient ring, much more compact SDP formulations are possible. This is usually the case when Gröbner bases for the ideal are available or easy to compute. The first case usually occurs in combinatorial optimization problems, and the latter one when only a few constraints are present (we illustrate this through an example in Section 7.1).

For concreteness, consider the problem of verifying a nonnegativity condition of a polynomial $f(x)$ on a set defined by equality constraints $g_i(x) = 0$ (i.e., an algebraic variety). Let $\{b_1, \ldots b_m\}$ be a *Gröbner basis* of the corresponding polynomial ideal $I := \langle g_i(x) \rangle$; see [8] for an excellent introduction

to computational algebra and Gröbner basis methods. It is easy to see then that the two statements

$$f(x) + \sum_i \lambda_i(x) g_i(x) \quad \text{is a sum of squares in } \mathbb{R}[x]$$

and

$$f(x) \quad \text{is a sum of squares in } \mathbb{R}[x]/I$$

are equivalent, and both are sufficient conditions for the nonnegativity of $f$ on the variety defined by $g_i(x) = 0$.

Therefore, we only need to look at sum of squares on quotient rings. This can be done by using exactly the same SDP techniques as in the standard case. There are two main differences:

- Instead of indexing the rows and columns of the matrix in the SDP by the usual monomials, we have to use the so-called *standard* monomials corresponding to the chosen Gröbner basis of the ideal $I$. These are the monomials which are not divisible by any leading term of the polynomials $b_i$ in the Gröbner basis.
- All operations are performed in the quotient ring, i.e., we take *normal form* of the terms after multiplication.

*Example 5.* Consider the problem of verifying if the polynomial $F = 10 - x^2 - y$ is nonnegative on the curve defined by $G := x^2 + y^2 - 1 = 0$ (the unit circle). According to our previous discussion, a natural starting point is to define the ideal $I = \langle G \rangle$, and to check whether $F$ is a sum of squares in $\mathbb{R}[x, y]/I$. We choose a simple graded lex monomial ordering. Therefore, we pick a partial basis of the quotient ring (here, we take only $\{1, x, y\}$) and write:

$$\begin{aligned} 10 - x^2 - y &= \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \\ &= q_{11} + q_{22}x^2 + q_{33}y^2 + 2q_{12}x \\ &\quad + 2q_{13}y + 2q_{23}xy \\ &\equiv (q_{11} + q_{33}) + (q_{22} - q_{33})x^2 + 2q_{12}x \\ &\quad + 2q_{13}y + 2q_{23}xy \qquad \mod I. \end{aligned}$$

Equating coefficients, we obtain again a simple SDP. Solving, we have:

$$Q = \begin{bmatrix} 9 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} = L^T L, \quad L = \frac{1}{\sqrt{2}} \begin{bmatrix} 3 & 0 & -\frac{1}{6} \\ 0 & 0 & \frac{\sqrt{35}}{6} \end{bmatrix},$$

and therefore

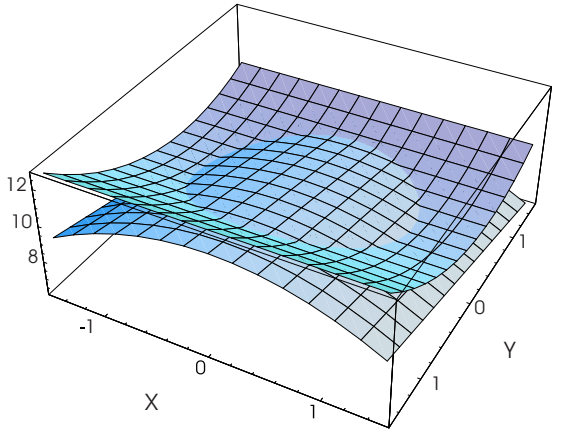



**Fig. 1.** The polynomials $F$ and $(3-\frac{y}{6})^2+\frac{35}{36}y^2$ take exactly the same values on the unit circle $x^2+y^2=1$. Thus, $F$ is nonnegative on the feasible set.

$$10-x^2-y \equiv (3-\frac{y}{6})^2+\frac{35}{36}y^2 \qquad \mod I,$$

which shows that $F$ is indeed SOS on $\mathbb{R}[x,y]/I$. A simple geometric interpretation is shown in Figure 1. By the condition above, the polynomial $F$ coincides with a SOS on the set defined by $G=0$, and thus it is obviously nonnegative on the variety.

Even though in the worst case Gröbner bases can be computationally troublesome, for many practical problems they are often directly available, or relatively easy to compute. A typical example is the case of combinatorial optimization problems, where the equations defining the Boolean ideal $x_i^2-1=0$ are already a Gröbner basis. Another frequent situation is when the ideal is defined by just one equality constraint, in which case the defining equation is again obviously a Gröbner basis of the corresponding ideal.

An advantage of the ideal-theoretic formulation is the ease by which structural results can be obtained through basic algebraic notions. For instance, in [18] it is shown that a certain finite convergence property holds for *all* zero dimensional radical ideals, generalizing earlier results by Lasserre [13] on boolean programming.

## 5 Symmetries

Yet another useful property that can be exploited in the SOS/SDP context is the presence of structural symmetries. While in some cases this mirrors the

underlying structure of existing physical systems, these features can also arise as a result of the chosen mathematical abstraction. In this regard, symmetry reduction techniques have been explored in several different contexts related to control theory, with dynamical systems [11] and geometric mechanics [14] being two prominent examples.

In our earlier work [10], these symmetry ideas are explored in the SOS/SDP framework using a dual approach that combines group representation and invariant theory. Here we focus on the former class of methods, and sketch the details in the remainder of this section. A particularly interesting application is the optimization of symmetric polynomials via SOS methods. In [17, 21], and based on [24], an SDP-based algorithm for the computation of a global lower bound for polynomial functions is described. When the polynomial to be minimized has certain invariance properties, then the theory described here can be fruitfully applied.

There are several advantages in exploiting symmetries:

- *Problem size.* The first immediate advantage is the reduction in problem size, as the new instance can have a significantly smaller number of variables and constraints.
- *Degeneracy removal.* In symmetric SDP problems, there are repeated eigenvalues of high multiplicity, that are difficult to handle numerically. These can be removed by a proper handling of the symmetry.
- *Conditioning and reliability.* Symmetry-aware methodologies have in general much better numerical conditioning, and the resulting smaller size instances are usually less prone to numerical errors.

The starting point in [10] is the definition of a general class of SDPs that are invariant under the action of a symmetry group. As a direct consequence of convexity, it is shown there that the solution of the SDP can always be assumed to lie in the *fixed point subspace* of the group action. Using Schur's lemma of representation theory, it is possible to show that in the appropriate symmetry-adapted basis, the matrices in the fixed-point subspace will have a block-diagonal structure. This reduces the original problem to a collection of smaller coupled SDPs, each block corresponding to an "isotypic component," and cardinality equal to the number of irreducible representations of the group that appear nontrivially. This allows for a notable reduction in both the number of decision variables and the size of the SDPs to be solved.

*Example 6.* Again, we illustrate the techniques with a specific example, this time taken from [10, Example 5.4]. This polynomial has an interesting dihedral symmetry, and is given by:

$$r(x, y) = x^6 + y^6 - x^4y^2 - y^4x^2 - x^4 - y^4 - x^2 - y^2 + 3x^2y^2 + 1. \quad (5)$$

This polynomial has the symmetry of the 8-element dihedral group $D_4$, with the actions on $\mathbb{R}^2$ generated by:

$$d := (x, y) \to (-y, x), \qquad s := (x, y) \to (y, x).$$

We are interested in finding a SOS decomposition of $r$. Using the standard approach discussed earlier, we would need to solve an SDP of dimensions $10 \times 10$. Using the methods discussed, a *symmetry-adapted basis* for the isotypic components can now be obtained, obtaining the corresponding basis vectors:

$$\begin{aligned} B_1 &= \{1, x^2 + y^2\} & B_4 &= \{x^2 - y^2\} \\ B_2 &= \emptyset & B_5^1 &= \{x, x^3, xy^2\} \\ B_3 &= \{xy\} & B_5^2 &= \{y, y^3, yx^2\}. \end{aligned}$$

Using this basis, and after symmetry reduction, the resulting SDPs are much simpler:

$$\begin{aligned} \begin{bmatrix} 1 & c_1 \\ c_1 & c_2 \end{bmatrix} &\geq 0 \\ 1 - 4c_2 - 4c_3 - 4c_4 &\geq 0 \\ -1 - c_2 - 2c_3 &\geq 0 \\ \begin{bmatrix} -1 - 2c_1 & c_3 & c_4 \\ c_3 & 1 & c_5 \\ c_4 & c_5 & -1 - 2c_5 \end{bmatrix} &\geq 0, \end{aligned}$$

so we have reduced the problem from one $10 \times 10$ SDP to four coupled smaller ones, of dimensions $2, 1, 1, 3$ respectively, which are considerably easier to solve. Solving these small SDPs, it is easy to obtain the final decomposition:

$$\begin{aligned} r(x, y) = &\frac{3825}{4096} + \left(\frac{x^2 + y^2}{2} - \frac{89}{64}\right)^2 \\ &+ \left(x^3 - y^2 x - \frac{5}{8}x\right)^2 + \left(y^3 - x^2 y - \frac{5}{8}y\right)^2. \end{aligned}$$

## 6 Combination of Techniques

A most appealing feature of the techniques described is the large extent to which they are mutually compatible. For instance, it is possible to combine very successfully sparsity and symmetry reduction techniques. For a particular polynomial arising from a geometric theorem proving problem, for instance, this combined use allowed the reduction from a very difficult $1001 \times 1001$ SDP, to a collection of fourteen coupled SDPs of dimensions ranging from $2 \times 2$ to $11 \times 11$), much more manageable computationally [20].

**Table 1.** Algebraic structures and SOS properties

| Standard | Equality constraints | Symmetries |
|---|---|---|
| polynomial ring $\mathbb{R}[x]$ | quotient ring $\mathbb{R}[x]/I$ | invariant ring $\mathbb{R}[x]^G$ |
| monomials ($\deg \leq k$) | *standard* monomials | isotypic components |
| $\frac{1}{(1-\lambda)^n} = \sum_{k=0}^{\infty} \binom{n+k-1}{k} \cdot \lambda^k$ | Hilbert series | Molien series |
| | Finite convergence on zero dimensional ideals | Block diagonalization |

In Table 1 we present a summary of the different concepts and techniques discussed earlier. Several of these complexity reduction techniques are already included in the currently available version of SOSTOOLS [22], and the rest will follow soon.

## 7 Examples

In this section we present an example of a control-oriented problem, where several of the techniques are applied.

### 7.1 Domain of Attraction of a Lyapunov Function

This problem has been analyzed in [17, Section 7.3], where it has been shown how use sum of squares techniques in the computation of positively invariant subsets.

Specifically, the problem analyzed there was to find bounds on the largest sublevel set of a Lyapunov function $V$ that is positively invariant. This can be done by solving the optimization problem:

$$\gamma_0 := \inf_{x,y \in \mathbb{R}^n} V(x,y) \qquad \text{subject to } \begin{cases} \dot{V}(x,y) = 0 \\ (x,y) \neq (0,0) \end{cases} \tag{6}$$

Since the minimization problem is constrained to the algebraic variety defined by the single equation $\dot{V} = 0$, we can easily work in the corresponding quotient ring. Let's see this in the concrete example analyzed in [17], taken from [25, Example S7].

*Example 7.* Consider the vector field given by:

$$\begin{aligned} \dot{x} &= -x + y \\ \dot{y} &= 0.1x - 2y - x^2 - 0.1x^3 \end{aligned}$$

and the Lyapunov function $V(x,y) := x^2 + y^2$. The system has three fixed points, at $(0,0), (-5 \pm \sqrt{6}, -5 \pm \sqrt{6})$.

A Gröbner basis of the ideal $I$ generated by $\dot{V} = 0$ is given by the single polynomial $\{10x^2 - 11xy + 20y^2 + 10yx^2 + yx^3\}$, which is just a scaled version of $\dot{V}$. The leading monomial, using a graded lex monomial ordering, is $yx^3$, so a basis for the quotient ring is given by the monomials $\{x^k, y^k, xy^k, x^2y^k,\ k \geq 0\}$.

To obtain a bound on the minimum of (6), we can find the largest value of $\gamma$ that satisfies

$$(V(x,y) - \gamma)(x^2 + y^2) \quad \text{is a sum of squares in } \mathbb{R}[x,y]/I.$$

Any value of $\gamma$ verifying the condition above gives a lower bound on $\gamma_0$, as is clear by evaluating the expression on any candidate feasible solution.

Taking as a partial basis of the quotient ring the monomials $\{x, y, x^2, xy, y^2, yx^2\}$, we obtain a $6 \times 6$ SDP that provides the exact answer, namely $\gamma \approx 7.111$.

## 8 Conclusions

Sum of squares based techniques, while still at the early stages of their development, have already shown an unprecedented flexibility and strength in solving many interesting problems in systems and control theory and related fields.

For their continuing success beyond the many current practical applications, it will be necessary to extend the size of the problems that can be reliably solved. While encouraging progress is continuously being made on the SDP solvers front (particularly with alternatives to interior-point methods), there is much to be gained from understanding their rich underlying algebraic structure. It is our hope that the results presented here make a convincing argument for the ideal suitability of the language and tools of computational algebra for this task.

## References

1. B. D. O. Anderson, N. K. Bose, and E. I. Jury (1975). Output feedback stabilization and related problems—solution via decision methods. IEEE Transactions on Automatic Control, 20:53–66.
2. S. Basu, R. Pollack, and M.-F. Roy (2003). Algorithms in real algebraic geometry. Springer-Verlag.
3. J. Bochnak, M. Coste, and M.-F. Roy (1998). Real Algebraic Geometry. Springer-Verlag.
4. S. Boyd and C. Barratt (1991). Linear Controller Design: Limits of Performance. Prentice-Hall.

5. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory. Volume 15 of Studies in Applied Mathematics. SIAM, Philadelphia, PA.
6. G. Chesi, A. Garulli, A. Tesi, and A. Vicino (2001). LMI-based techniques for solving quadratic distance problems. Proc. IEEE Conference on Decision and Control, 3587–3592.
7. M. D. Choi, T. Y. Lam, and B. Reznick (1995). Sums of squares of real polynomials. Proc. Symposia in Pure Mathematics, 58(2):103–126.
8. D. A. Cox, J. B. Little, and D. O'Shea (1997). Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. Springer-Verlag.
9. P. Dorato, W. Yang, and C. Abdallah (1997). Robust multi-objective feedback design by quantifier elimination. J. Symbolic Computation, 24:153–159.
10. K. Gatermann and P. A. Parrilo (2004). Symmetry groups, semidefinite programs, and sums of squares. Journal of Pure and Applied Algebra, 92:95–128.
11. M. Golubitsky, I. Stewart, and D. G. Schaeffer (1988). Singularities and Groups in Bifurcation Theory II, Volume 69 of Applied Mathematical Sciences, Springer-Verlag, NY.
12. M. Jirstrand (1997). Nonlinear control system design by quantifier elimination. J. Symbolic Computation, 24:137–152.
13. J. B. Lasserre (2001). Global optimization with polynomials and the problem of moments. SIAM J. Optim. 11(3):796–817.
14. J. E. Marsden and T. Ratiu (1999). Introduction to Mechanics and Symmetry. Volume 17 of Texts in Applied Mathematics. Springer-Verlag, 2nd edition.
15. B. Mishra (1993). Algorithmic Algebra. Springer-Verlag.
16. Y. Nesterov (2000). Squared functional systems and optimization problems. In J. Frenk, C. Roos, T. Terlaky, and S. Zhang (Editors). High Performance Optimization, 405–440. Kluwer Academic Publishers.
17. P. A. Parrilo (2000). Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology. Available at `resolver.caltech.edu/CaltechETD:etd-05062004-055516`.
18. P. A. Parrilo (2002). An explicit construction of distinguished representations of polynomials nonnegative over finite sets. Technical Report IfA Technical Report AUT02-02, ETH Zürich. Available from `control.ee.ethz.ch/~parrilo`.
19. P. A. Parrilo (2003). Semidefinite programming relaxations for semialgebraic problems. Math. Prog., 96(2, Ser. B):293–320.
20. P. A. Parrilo and R. Peretz (2004). An inequality for circle packings proved by semidefinite programming. Discrete and Computational Geometry, 31(3):357–367.
21. P. A. Parrilo and B. Sturmfels (2003). Minimizing polynomial functions. In S. Basu and L. González-Vega (Editors). Algorithmic and Quantitative Real Algebraic Geometry. Volume 60 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. AMS. Available from `arXiv:math.OC/0103170`.
22. S. Prajna, A. Papachristodoulou, and P. A. Parrilo (2002). SOSTOOLS: Sum of squares optimization toolbox for MATLAB. Available from `www.cds.caltech.edu/sostools` and `control.ee.ethz.ch/~parrilo/sostools`.
23. B. Reznick (1978). Extremal PSD forms with few terms. Duke Mathematical Journal, 45(2):363–374.

24. N. Z. Shor (1987). Class of global minimum bounds of polynomial functions. Cybernetics, 23(6):731–734, 1987. Russian orig.: Kibernetika, 6:9–11, 1987.
25. A. Tesi, F. Villoresi, and R. Genesio (1996). On the stability domain estimation via a quadratic Lyapunov function: convexity and optimality properties for polynomial systems. IEEE Transactions on Automatic Control, 41(11):1650–1657.

# Interior-Point Algorithms for Semidefinite Programming Problems Derived from the KYP Lemma

Lieven Vandenberghe[1], V. Ragu Balakrishnan[2], Ragnar Wallin[3], Anders Hansson[3], and Tae Roh[1]

[1] Department of Electrical Engineering, University of California, Los Angeles. vandenbe@ee.ucla.edu, roh@ee.ucla.edu
[2] School of Electrical and Computer Engineering, Purdue University. ragu@ecn.purdue.edu
[3] Division of Automatic Control, Department of Electrical Engineering, Linköping University. ragnarw@isy.liu.se, hansson@isy.liu.se

We discuss fast implementations of primal-dual interior-point methods for semidefinite programs derived from the Kalman-Yakubovich-Popov lemma, a class of problems that are widely encountered in control and signal processing applications. By exploiting problem structure we achieve a reduction of the complexity by several orders of magnitude compared to general-purpose semidefinite programming solvers.

## 1 Introduction

We discuss efficient implementations of interior-point methods for semidefinite programming problems (SDPs) of the form

$$\begin{array}{ll} \text{minimize} & q^T x + \sum_{k=1}^{L} \mathbf{Tr}(Q_k P_k) \\ \text{subject to} & \begin{bmatrix} A_k^T P_k + P_k A_k & P_k B_k \\ B_k^T P_k & 0 \end{bmatrix} + \displaystyle\sum_{i=1}^{p} x_i M_{ki} \succeq N_k, \quad k = 1, \ldots, L. \end{array} \tag{1}$$

The optimization variables are $x \in \mathbf{R}^p$ and $L$ matrices $P_k \in \mathbf{S}^{n_k}$, where $\mathbf{S}^n$ denotes the space of symmetric matrices of dimension $n \times n$. The problem data are $q \in \mathbf{R}^p$, $Q_k \in \mathbf{S}^{n_k}$, $A_k \in \mathbf{R}^{n_k \times n_k}$, $B_k \in \mathbf{R}^{n_k \times m_k}$, $M_{ki} \in \mathbf{S}^{n_k+m_k}$, and $N_k \in \mathbf{S}^{n_k+m_k}$. If $n_k = 0$, the $k$th constraint is interpreted as the linear matrix inequality (LMI) $\sum_{i=1}^{p} x_i M_{ki} \succeq N_k$. The SDPs we study can therefore include arbitrary LMI constraints. At the end of this section we will list several assumptions made about the problem data. The most important of these assumptions is that $(A_k, B_k)$ is controllable for $k = 1, \ldots, L$.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 195–238, 2005.

We refer to SDPs of the form (1) as KYP-SDPs, and to the constraints in the problem as KYP-LMIs, for the following reason. The Kalman-Yakubovich-Popov (KYP) lemma states that the semi-infinite frequency domain inequality

$$\left[\begin{array}{c}(j\omega I - A)^{-1}B\\ I\end{array}\right]^* M \left[\begin{array}{c}(j\omega I - A)^{-1}B\\ I\end{array}\right] \succ 0, \qquad \omega \in \mathbf{R}, \tag{2}$$

where $A \in \mathbf{R}^{n\times n}$ does not have imaginary eigenvalues, holds if and only if the strict LMI

$$\left[\begin{array}{cc}A^TP + PA & PB\\ B^TP & 0\end{array}\right] + M \succ 0$$

with variable $P \in \mathbf{S}^n$ is feasible. Moreover, if $(A, B)$ is controllable, then the nonstrict frequency domain inequality

$$\left[\begin{array}{c}(j\omega I - A)^{-1}B\\ I\end{array}\right]^* M \left[\begin{array}{c}(j\omega I - A)^{-1}B\\ I\end{array}\right] \succeq 0, \qquad \omega \in \mathbf{R}, \tag{3}$$

holds if and only if the nonstrict LMI

$$\left[\begin{array}{cc}A^TP + PA & PB\\ B^TP & 0\end{array}\right] + M \succeq 0 \tag{4}$$

is feasible (for a discussion of these results from a semidefinite programming duality perspective, see [11]). The KYP lemma forms the basis of some of the most important applications of SDPs in control; see, for example, [8, 42, 26, 39, 38, 31, 12, 29].

The constraints in the KYP-SDP (1) have the same general form as (4), with $M$ replaced with an affine function of the optimization variable $x$. If $Q_k = 0$, the KYP-SDP is therefore equivalent to the semi-infinite SDP

$$\begin{array}{ll}\text{min.} & q^Tx\\ \text{s.t.} & \left[\begin{array}{c}(j\omega I - A_k)^{-1}B_k\\ I\end{array}\right]^* (\mathcal{M}_k(x) - N_k) \left[\begin{array}{c}(j\omega I - A_k)^{-1}B_k\\ I\end{array}\right] \succeq 0,\\ & k = 1, \ldots, L,\end{array} \tag{5}$$

with variable $x$, where $\mathcal{M}_k(x) = \sum_{i=1}^p x_i M_{ki}$. More details and examples, including some applications in which $Q_k \neq 0$, are given in §2.

KYP-SDPs are difficult to solve using general-purpose SDP software packages [44, 47, 3, 18, 9, 13, 24, 48]. The difficulty stems from the very high number of optimization variables ($p + \sum_k n_k(n_k+1)/2$). Even moderate values of $n_k$ (say, a few hundred) result in very large scale SDPs, with several 10,000 or 100,000 variables. This is unfortunate, because in many applications the variables $P_k$ are of little intrinsic interest. They are introduced as auxiliary variables, in order to convert the semi-infinite frequency-domain constraint (3) into a finite-dimensional LMI (4).

For this reason, several researchers have proposed alternatives to standard interior-point methods for solving KYP-SDPs. These methods include cutting-plane methods (such as the analytic center cutting-plane method) [41, 32, 33, 34, 25], interior-point methods based on alternative barrier functions for the frequency-domain constraint [32], and interior-point methods combined with conjugate gradients [28, 29, 52, 20].

In this paper we examine the possibility of exploiting KYP-SDP problem structure to speed up standard primal-dual interior-point methods of the type used in state-of-the-art solvers like SeDuMi [44, 45] and SDPT3 [47]. Straightforward linear algebra techniques will allow us to implement the same interior-point methods at a cost that is orders of magnitude less than the cost of general-purpose implementations. More specifically, if $n_k = n$, $m_k = 1$ for $k = 1, \ldots, L$, and $p = O(n)$, then the cost per iteration of a general-purpose solver grows at least as $n^6$ as a function of $n$. Exploiting structure will allow us to reduce the complexity per iteration to $n^3$. Similar results have previously been obtained for dual barrier methods applied to special classes of KYP-SDPs, for example, KYP-SDPs derived for discrete-time FIR systems [6, 22, 25]. The results in this paper can be viewed as an extension of these techniques to general KYP-SDPs, and to primal-dual interior-point methods.

### Outline of the Paper

The paper is organized as follows. In §2 we give an overview of applications, illustrating that KYP-SDPs are widely encountered in control. In §3 we present some basic facts about SDPs, SDP duality, and primal-dual interior-point methods for solving them. In §4 we explain in more detail the computations involved in solving KYP-SDPs using general-purpose software, and justify our estimate of an order $n^6$ complexity per iteration. We also describe a dual reformulation of the KYP-SDP which can be solved at a cost of roughly $O(n^4)$ per iteration, using general-purpose software. In §5 we describe techniques that exploit additional problem structure and result in a complexity of roughly $O(n^3)$ per iteration, for either the primal or dual formulation. The relation between the methods discussed in §4 and §5 is illustrated in Table 1. The results of some numerical experiments are described in §6. In §7 we discuss extensions of the techniques in §4 and §5, to problems with multiple constraints

**Table 1.** Relation between the methods in §4 and §5, and estimates of their complexity per iteration (for KYP-SDPs with $L = 1$, $n_1 = n$, $m_1 = 1$, $p = O(n)$).

| | Primal formulation | Dual formulation |
|---|---|---|
| General-purpose | $O(n^6)$ (§4.1) | $O(n^4)$ (§4.2) |
| Special-purpose | $O(n^3)$ (§5) | $O(n^3)$ (§5) |

($L > 1$), and KYP-LMIs for multi-input systems ($m_k > 1$). Conclusions and some suggestions for future research are presented in §8.

The paper also includes several appendices. Appendix A provides additional background on semidefinite programming, and a detailed summary of the primal-dual interior-point of [46]. The other appendices contain proofs of results in the paper, and discussion of relaxed assumptions.

### Assumptions

We will assume that the pairs $(A_k, B_k)$, $k = 1, \ldots, L$, are controllable. Controllability implies that the linear mappings $\mathcal{K}_k : \mathbf{S}^{n_k} \to \mathbf{S}^{n_k+m_k}$, defined by

$$\mathcal{K}_k(P) = \begin{bmatrix} A_k^T P + P A_k & P B_k \\ B^T P & 0 \end{bmatrix},$$

have full rank (see §4.2). In addition, we assume that the matrices $M_{ki}$ are such that the mapping

$$(P_1, P_2, \ldots, P_L, x) \quad \mapsto \quad \mathbf{diag}(\mathcal{K}_1(P_1) + \mathcal{M}_1(x), \ldots, \mathcal{K}_L(P_L) + \mathcal{M}_L(x)) \quad (6)$$

has full rank, where $\mathcal{M}_k(x) = \sum_{i=1}^{p} x_i M_{ki}$. In other words, the lefthand sides of the constraints in (1) are zero if and only if $P_k = 0$, $k = 1, \ldots, L$, and $x = 0$.

In fact these two assumptions can be relaxed. Controllability of $(A_k, B_k)$ can be replaced with stabilizability, provided the range of $Q_k$ is in the controllable subspace of $(A_k, B_k)$; see Appendix D. Moreover, a problem for which (6) does not have full rank, can always be converted to an equivalent reduced order problem for which the full rank assumption holds; see Appendix E.

Throughout the paper we assume that the problem data and parameters are real. The generalization to complex data should be straightforward.

### Notation

The space of symmetric $l \times l$ matrices is denoted $\mathbf{S}^l$. For $X \in \mathbf{S}^l$, $\mathbf{svec}(X)$ denotes the $l(l+1)/2$ vector containing the lower triangular elements of $X$:

$$\mathbf{svec}(X) = (x_{11}, x_{21}, \ldots, x_{l1}, x_{22}, \ldots, x_{l2}, \ldots, x_{l-1,l-1}, x_{l,l-1}, x_{ll}).$$

The space of symmetric block-diagonal matrices with block dimensions $l_1$, $\ldots$, $l_L$ is denoted $\mathbf{S}^{l_1} \times \mathbf{S}^{l_2} \times \cdots \times \mathbf{S}^{l_L}$. If $X_1 \in \mathbf{S}^{l_1}$, $\ldots$, $X_L \in \mathbf{S}^{l_L}$, then $\mathbf{diag}(X_1, \ldots, X_L)$ denotes the block-diagonal matrix with $X_1$, $\ldots$, $X_L$ as its diagonal blocks.

The space of Hermitian $l \times l$ matrices is denoted $\mathbf{H}^l$. For $A \in \mathbf{S}^l$ ($A \in \mathbf{H}^l$), $A \succeq 0$ means $A$ is positive semidefinite, and the set of positive semidefinite symmetric (Hermitian) matrices of dimension $l$ is denoted $\mathbf{S}^l_+$ ($\mathbf{H}^l_+$). Similarly,

$A \succ 0$ means $A$ is positive definite; $\mathbf{S}^l_{++}$ and $\mathbf{H}^l_{++}$ are the sets of positive definite symmetric, resp. Hermitian, matrices.

The Hadamard (componentwise) product $A \circ B$ of two matrices $A$, $B$ of equal dimensions is defined by $(A \circ B)_{ij} = a_{ij}b_{ij}$. The $i$th unit vector is denoted $e_i$.

## 2 Applications of KYP-SDPs

While the form of the KYP-SDP (1) and the KYP-LMIs (2) and (3) may appear very special, they are widely encountered in control and signal processing. We give a representative list of applications along with a brief description.

### 2.1 Optimization Problems with Frequency-Domain Inequalities

As we already noted, a KYP-SDP (1) with an objective that does not depend on the variables $P_k$ (*i.e.*, $Q_k = 0$), is equivalent to an optimization problem of the form (5), in which we minimize a linear cost function subject to frequency-domain inequalities (FDIs) of the form

$$H_k(\omega, x) \succeq 0 \qquad \omega \in \mathbf{R}. \tag{7}$$

Here $H_k : \mathbf{R} \times \mathbf{R}^p \to \mathbf{H}^m$ is defined as

$$H_k(\omega, x) = \begin{bmatrix} (j\omega I - A_k)^{-1}B_k \\ I \end{bmatrix}^* (\mathcal{M}_k(x) - N_k) \begin{bmatrix} (j\omega I - A_k)^{-1}B_k \\ I \end{bmatrix}.$$

Below we list a number of applications of problems with FDI constraints. It is important to note that in these applications, $x$ is usually the design variable that we are interested in; the matrix $P$ in the SDP formulation is an auxiliary variable, introduced to represent an infinite family of inequalities (7) as a single matrix inequality.

#### Linear System Analysis and Design

A well-known convex reformulation of the problem of linear time-invariant (LTI) controller design for LTI systems is via the Youla parametrization; see for example [7]. The underlying optimization problem here is to find $x$ such that

$$T(s, x) = T_1(s) + T_2(s) \left( \sum_{i=1}^{p} x_i Q_i(s) \right) T_3(s),$$

satisfies a number of affine inequalities for $s = j\mathbf{R}$, where $T_i$ and $Q_i$ are given stable rational transfer function matrices [7, 26, 39]. These inequalities are readily expressed as FDIs of the form (7).

**Digital Filter Design**

This application involves the discrete-time version of the FDI (7). The standard digital filter design problem consists of designing

$$T(z,x) = \sum_{i=1}^{p} x_i T_i(z),$$

where $T_i : \mathbf{C} \to \mathbf{C}$ are given transfer functions, and $x$ is to be determined so that $G(z,x)$ satisfies certain constraints. When $T_i(z) = z^{-i}$, we have a finite-impulse response (FIR) design problem. The constraints can be *magnitude constraints* of the form

$$|T(e^{j\theta},x)| \leq U(e^{j\theta}), \qquad \theta \in [0,2\pi), \tag{8}$$

or *phase constraints*

$$\angle T(e^{j\theta},x) \leq R(e^{j\theta}), \qquad \theta \in [0,2\pi), \tag{9}$$

Extensions where $T_i$ are more general filter banks, and when $T_i$ are matrix-valued transfer functions are immediate [10]. Other variations include optimal array pattern synthesis [51].

When $U(e^{j\theta})$ and $\tan(R(e^{j\theta}))$ are given (or can be approximated) as rational functions of $e^{j\theta}$, it is straightforward to express constraints (8) as the unit-circle counterparts of inequalities (7).

Other types of filter design problems include two-sided magnitude constraints

$$L(e^{j\theta}) \leq |T(e^{j\theta},x)| \leq U(e^{j\theta}), \qquad \theta \in [0,2\pi),$$

and no phase constraints. These constraints can be expressed as linear FDIs via a change of variables; see [50, 5, 6, 14, 21].

**Robust Control Analysis Using Integral Quadratic Constraints**

Robust control [54, 23] deals with the analysis of and design for control system models that incorporate uncertainties explicitly in them. A sufficient condition for robust stability (*i.e.*, stability of the model irrespective of the uncertainties) can be unified in the framework of integral quadratic constraints (IQCs). The numerical problem underlying the IQC-based robust stability conditions is the following [38, 31, 12]: Find $x \in \mathbf{R}^m$ such that for $\epsilon > 0$ and for all $\omega \in \mathbf{R}$,

$$\begin{bmatrix} T(j\omega) \\ I \end{bmatrix}^* \Pi(j\omega,x) \begin{bmatrix} T(j\omega) \\ I \end{bmatrix} \preceq -2\epsilon I, \tag{10}$$

where $T : \mathbf{C} \to \mathbf{C}^{m\times m}$ is a given real-rational function, and $\Pi : \mathbf{C} \times \mathbf{R}^p \to \mathbf{C}^{2m\times 2m}$ is a linear function of $x$ for fixed $\omega$, and is a real-rational function of $\omega$ for fixed $x$. Clearly, (10) corresponds to a special instance of (3).

Multiple FDIs of the form (10) result with more sophisticated (and better) sufficient conditions for robust stability with the IQC framework; see for example [17, 30].

### 2.2 Linear-Quadratic Regulators

Consider the continuous-time dynamical system model

$$\dot{x} = Ax + Bu \tag{11}$$

with initial value $x(0) = x_0$, $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $x(t) \in \mathbf{R}^n$, and $u(t) \in \mathbf{R}^m$. Assume that $(A, B)$ is controllable.

#### Riccati Equations

Define the cost index

$$J = \int_0^\infty \begin{bmatrix} x \\ u \end{bmatrix}^T M \begin{bmatrix} x \\ u \end{bmatrix} dt \tag{12}$$

where

$$M = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \in \mathbf{S}^{n+m}$$

with $R \succ 0$. It is well-known (see, *e.g*, [53]), that the infimal value of $J$ with respect to $u(\cdot)$ subject to (11) and such that $\lim_{t\to\infty} x(t) = 0$ is, whenever it exists, given by $x_0^T P x_0$, where $P \in \mathbf{S}^n$ solves the KYP-SDP

$$\begin{array}{ll} \text{maximize} & x_0^T P x_0 \\ \text{subject to} & \begin{bmatrix} A^T P + PA & PB \\ B^T P & 0 \end{bmatrix} + M \succeq 0. \end{array} \tag{13}$$

The optimal $u(\cdot)$ is given as a state feedback $u(t) = -R^{-1}(PB + S)^T x(t)$. Here we see an application where the variable $P$ is of intrinsic interest and appears in the objective. For this special case, of course, the optimal $P$ can be found by solving an algebraic Riccati equation

$$A^T P + PA + Q - (PB + S)^T R^{-1}(PB + S) = 0,$$

and very efficient methods based on the real ordered Schur form of an associated matrix pencil are available. The computational complexity of these methods is in the order of $n^3$. However, numerical experience have shown that for certain ill-conditioned algebraic Riccati equations the KYP-SDP-formulation is not ill-conditioned. In some cases it can therefore be beneficial to solve algebraic Riccati equations via the SDP formulation. Moreover, slight generalizations of the above problem formulation require the solution of general KYP-SDPs. An example is given next.

#### Quadratic Constraints

Define the cost indices

$$J_i = \int_0^\infty \begin{bmatrix} x \\ u \end{bmatrix}^T M_i \begin{bmatrix} x \\ u \end{bmatrix} dt, \qquad i = 0, \ldots, p \tag{14}$$

where $M_i \in \mathbf{S}^{n+m}$. Consider the constrained optimization problem

$$\begin{array}{ll} \text{minimize} & J_0 \\ \text{subject to} & J_i \leq c_i, \quad i = 1, \ldots, p \\ & (11) \text{ and } \lim_{t\to\infty} x(t) = 0 \end{array} \tag{15}$$

with respect to $u(\cdot)$. The optimal value to this problem, whenever it exists, is given by $x_0^T P x_0$, where $P$ solves the KYP-SDP

$$\begin{array}{ll} \text{maximize} & x_0^T P x_0 - c^T x \\ \text{subject to} & \begin{bmatrix} A^T P + PA & PB \\ B^T P & 0 \end{bmatrix} + M_0 + \sum_{i=1}^p x_i M_i \succeq 0 \\ & x_i \geq 0, \quad i = 1, \ldots, p \end{array} \tag{16}$$

(see [8, page 151]). The optimal $u(\cdot)$ is given as a state feedback $u(t) = -R^\dagger (PB + S)^T x(t)$, where

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = M_0 + \sum_{i=1}^p x_i M_i.$$

Here we see an application where both the variables $P$ and $x$ are of intrinsic interest. Moreover, we have multiple constraints, some of which only involve $x$.

### 2.3 Quadratic Lyapunov Function Search

Consider the continuous-time dynamical system model

$$\dot{x} = f(x, u, w, t), \qquad z = g(x, u, w, t), \qquad y = h(x, u, w, t) \tag{17}$$

where $x : \mathbf{R}_+ \to \mathbf{R}^n$, $u : \mathbf{R}_+ \to \mathbf{R}^{n_u}$, $w : \mathbf{R}_+ \to \mathbf{R}^{n_w}$, $z : \mathbf{R}_+ \to \mathbf{R}^{n_z}$, and $y : \mathbf{R}_+ \to \mathbf{R}^{n_y}$. $x$ is referred to as the state, $u$ is the control input, $w$ is the exogenous input, $z$ is the output of interest and $y$ is the measured output. Models such as (17) are ubiquitous in engineering. (We have presented a continuous-time model only for convenience; the statements we make are equally applicable to discrete-time models.)

A powerful tool for the analysis of and design for model (17) proceeds via the use of quadratic Lyapunov functions. Suppose that for some $P \in \mathbf{S}^n_{++}$, the function $V(\psi) \stackrel{\Delta}{=} \psi^T P \psi$ satisfies

$$\frac{d}{dt} V(x, t) < 0 \text{ along the trajectories of (17)}, \tag{18}$$

then all trajectories of model (17) go to zero. For a number of special instances of system (17), the numerical search for Lyapunov functions results in feasibility problems with KYP-LMI constraints; see, for example, [8]. As an example, consider the system

$$\dot{x} = Ax + B_p p, \qquad q = C_q x + D_{qp} p, \qquad p = \Delta(t) q, \qquad \|\Delta(t)\| \leq 1, \quad (19)$$

where $\Delta : \mathbf{R}_+ \to \mathbf{R}^{m \times m}$. The existence of a quadratic Lyapunov function such that $dV(x,t)/dt < 0$ holds along the trajectories of (19) is equivalent to the following KYP-LMI:

$$P \succ 0, \qquad \begin{bmatrix} A^T P + PA + C_q^T C_q & PB_p + C_q^T D_{qp} \\ (PB_p + C_q^T D_{qp})^T & -(I - D_{qp}^T D_{qp}) \end{bmatrix} \prec 0. \quad (20)$$

If $(A, C_q)$ is observable, the inequality $P \succ 0$ is implied by the second LMI, which is a (strict) KYP-LMI.

Variations of this basic idea underlie a very long list of recent results in systems and control theory that lead to KYP LMIs; the following list is by no means comprehensive:

- Robust stability of norm-bound systems with structured perturbations [15, 43, 8].
- Robust stability of parameter-dependent systems [19, 8].
- $\mathbf{H}_\infty$ controller synthesis [2].
- Gain-scheduled controller synthesis [40, 1, 49].

## 3 Interior-Point Algorithms for Semidefinite Programming

### 3.1 Semidefinite Programming

Let $\mathcal{V}$ be a finite-dimensional real vector space, with inner product $\langle u, v \rangle$. Let

$$\mathcal{A} : \mathcal{V} \to \mathbf{S}^{l_1} \times \mathbf{S}^{l_2} \times \cdots \times \mathbf{S}^{l_L}, \qquad \mathcal{B} : \mathcal{V} \to \mathbf{R}^r$$

be linear mappings, and suppose $c \in \mathcal{V}$, $D = \mathbf{diag}(D_1, D_2, \ldots, D_L) \in \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L}$, and $d \in \mathbf{R}^r$ are given. The optimization problem

$$\begin{array}{ll} \text{minimize} & \langle c, y \rangle \\ \text{subject to} & \mathcal{A}(y) + D \preceq 0 \\ & \mathcal{B}(y) + d = 0 \end{array} \quad (21)$$

with variable $y \in \mathcal{V}$ is called a *semidefinite programming problem* (SDP). The dual SDP associated with (21) is defined as

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(DZ) + d^T z \\ \text{subject to} & \mathcal{A}^{\text{adj}}(Z) + \mathcal{B}^{\text{adj}}(z) + c = 0 \\ & Z \succeq 0, \end{array} \tag{22}$$

where

$$\mathcal{A}^{\text{adj}} : \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L} \to \mathcal{V}, \qquad \mathcal{B}^{\text{adj}} : \mathbf{R}^r \to \mathcal{V}$$

denote the adjoints of $\mathcal{A}$ and $\mathcal{B}$. The variables in the dual problem are $Z \in \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L}$, and $z \in \mathbf{R}^r$. We refer to $Z$ as the dual variable (or multiplier) associated with the LMI constraint $\mathcal{A}(y) + D \preceq 0$, and to $z$ as the multiplier associated with the equality constraint $\mathcal{B}(y) + d = 0$.

### 3.2 Interior-Point Algorithms

Primal-dual interior-point methods solve the pair of SDPs (21) and (22) simultaneously. At each iteration they solve a set of linear equations of the form

$$-W\Delta ZW + \mathcal{A}(\Delta y) = R \tag{23}$$

$$\mathcal{A}^{\text{adj}}(\Delta Z) + \mathcal{B}^{\text{adj}}(\Delta z) = r_{\text{du}} \tag{24}$$

$$\mathcal{B}(\Delta y) = r_{\text{pri}}, \tag{25}$$

to compute primal and dual search directions $\Delta y \in \mathcal{V}$, $\Delta Z \in \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L}$, $\Delta z \in \mathbf{R}^r$. The scaling matrix $W$ and the righthand side $R$ in these equations are block-diagonal and symmetric ($W, R \in \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L}$), and $W$ is positive definite. The value of $W$, as well as the values of the righthand sides $R$, $r_{\text{du}}$, and $r_{\text{pri}}$, change at each iteration, and also depend on the particular algorithm used. We will call these equations *Newton equations* because they can be interpreted as a linearization of modified optimality conditions. We refer to appendix A, which gives a complete description of one particular primal-dual method, for more details. Primal or dual interior-point methods give rise to equations that have the same form as (23)–(25), with different definitions of $W$ and the righthand sides. In this paper we make no assumptions about $W$, other than positive definiteness, so our results apply to primal and dual methods as well.

Since in practice the number of iterations is roughly independent of problem size (and of the order of 10–50), the overall cost of solving the SDP is roughly proportional to the cost of solving a given set of equations of the form (23)–(25).

### 3.3 General-Purpose Solvers

In a general-purpose implementation of an interior-point method it is assumed that $\mathcal{V}$ is the Euclidean vector space $\mathbf{R}^s$ of dimension $s = \dim \mathcal{V}$, and that $\mathcal{A}$ and $\mathcal{B}$ are given in the canonical form

$$\mathcal{A}(y) = \sum_{i=1}^{s} y_i F_i, \qquad \mathcal{B}(y) = By.$$

The matrices $F_i \in \mathbf{S}^{l_1} \times \mathbf{S}^{l_2} \times \cdots \times \mathbf{S}^{l_L}$ and $B \in \mathbf{R}^{r \times s}$ are stored in a sparse matrix format.

The equations (23)–(25) are solved by eliminating $\Delta Z$ from the first equation, and substituting $\Delta Z = W^{-1}(\mathcal{A}(\Delta y) - R)W^{-1}$ in the second equation. This yields a symmetric indefinite set of linear equations in $\Delta y$, $\Delta z$:

$$\mathcal{A}^{\rm adj}(W^{-1}\mathcal{A}(\Delta y)W^{-1}) + \mathcal{B}^{\rm adj}(\Delta z) = r_{\rm du} + \mathcal{A}^{\rm adj}(W^{-1}RW^{-1}) \tag{26}$$

$$\mathcal{B}(\Delta y) = r_{\rm pri}. \tag{27}$$

Using the canonical representation of $\mathcal{A}$ and $\mathcal{B}$, these equations can be written as

$$\begin{bmatrix} H & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_{\rm du} + g \\ r_{\rm pri} \end{bmatrix},$$

where

$$H_{ij} = \mathbf{Tr}(F_i W^{-1} F_j W^{-1}), \quad i, j = 1, \ldots, s$$
$$g_i = \mathbf{Tr}(F_i W^{-1} R W^{-1}), \quad i = 1, \ldots, s.$$

If the SDP has no equality constraints, the equations reduce to

$$\mathcal{A}^{\rm adj}(W^{-1}\mathcal{A}(\Delta y)W^{-1}) = r_{\rm du} + \mathcal{A}^{\rm adj}(W^{-1}RW^{-1}). \tag{28}$$

*i.e.*,

$$H\Delta y = r_{\rm du} + g.$$

The matrix $H$ in this system is positive definite and almost always dense, so the cost of solving the equations is $(1/3)s^3$. This is only a lower bound on the actual cost per iteration, which also includes the cost of forming $H$. Even though sparsity in the matrices $F_i$ helps, the cost of constructing $H$ is often substantially higher than the cost of solving the equations.

## 4 General-Purpose SDP Solvers and KYP-SDPs

In this section we use the observations made in §3 to estimate the cost of solving KYP-SDPs with general-purpose interior-point software. For simplicity we assume that $L = 1$, $n_1 = n$, $m_1 = 1$, and consider the problem

$$\begin{array}{ll} \text{minimize} & q^T x + \mathbf{Tr}(QP) \\ \text{subject to} & \begin{bmatrix} A^T P + PA & PB \\ B^T P & 0 \end{bmatrix} + \sum_{i=1}^{p} x_i M_i \succeq N, \end{array} \tag{29}$$

where $A \in \mathbf{R}^{n \times n}$, $B \in \mathbf{R}^n$, with $(A, B)$ controllable. The extension to problems with multiple inputs ($m > 1$) and multiple constraints ($L > 1$) is discussed in §7.

In §4.1 we first make precise our earlier claim that the cost of a general-purpose solver applied to (1) grows at least as $n^6$, if $p = O(n)$. In §4.2 we then describe a straightforward technique, based on semidefinite programming duality, that reduces the cost to order $n^4$.

### 4.1 Primal Formulation

We can express the KYP-SDP (29) as

$$\begin{array}{ll} \text{minimize} & q^T x + \mathbf{Tr}(QP) \\ \text{subject to} & \mathcal{K}(P) + \mathcal{M}(x) \succeq N \end{array} \tag{30}$$

where

$$\mathcal{K}(P) = \begin{bmatrix} A^T P + PA & PB \\ B^T P & 0 \end{bmatrix}, \qquad \mathcal{M}(x) = \sum_{i=1}^{p} x_i M_i. \tag{31}$$

This is in the general form (21), with $\mathcal{V} = \mathbf{S}^n \times \mathbf{R}^p$, and

$$y = (P, x), \qquad c = (Q, q), \qquad D = N, \qquad \mathcal{A}(P, x) = -\mathcal{K}(P) - \mathcal{M}(x).$$

The adjoint of $\mathcal{A}$ is $\mathcal{A}^{\text{adj}}(Z) = -\left(\mathcal{K}^{\text{adj}}(Z), \mathcal{M}^{\text{adj}}(Z)\right)$, where

$$\mathcal{K}^{\text{adj}}(Z) = \begin{bmatrix} A & B \end{bmatrix} Z \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} I & 0 \end{bmatrix} Z \begin{bmatrix} A^T \\ B^T \end{bmatrix}, \qquad \mathcal{M}^{\text{adj}}(Z) = \begin{bmatrix} \mathbf{Tr}(M_1 Z) \\ \vdots \\ \mathbf{Tr}(M_p Z) \end{bmatrix}.$$

The dual problem of (32) is therefore

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(NZ) \\ \text{subject to} & \mathcal{K}^{\text{adj}}(Z) = Q, \quad \mathcal{M}^{\text{adj}}(Z) = q \\ & Z \succeq 0, \end{array} \tag{32}$$

with variable $Z \in \mathbf{S}^{n+1}$.

A general-purpose primal-dual method applied to (30) generates iterates $x$, $P$, $Z$. At each iteration it solves a set of linear equations of the form (23)–(25) with variables $\Delta x$, $\Delta P$, $\Delta Z$:

$$W \Delta Z W + \mathcal{K}(\Delta P) + \mathcal{M}(\Delta x) = R_1 \tag{33}$$

$$\mathcal{K}^{\text{adj}}(\Delta Z) = R_2 \tag{34}$$

$$\mathcal{M}^{\text{adj}}(\Delta Z) = r, \tag{35}$$

for some positive definite $W$ and righthand sides $R_1$, $R_2$, $r$. These equations are solved by eliminating $\Delta Z$, reducing them to a smaller positive definite system (28). The reduced equations can be written in matrix-vector form as

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{bmatrix} \begin{bmatrix} \mathbf{svec}(\Delta P) \\ \Delta x \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \tag{36}$$

The blocks of the coefficient matrix are defined by the identities

$$\begin{aligned} H_{11}\,\mathbf{svec}(\Delta P) &= \mathbf{svec}\left(\mathcal{K}^{\mathrm{adj}}(W^{-1}\mathcal{K}(\Delta P)W^{-1})\right) \\ H_{12}\Delta x &= \mathbf{svec}\left(\mathcal{K}^{\mathrm{adj}}(W^{-1}\mathcal{M}(\Delta x)W^{-1})\right) \\ H_{22}\Delta x &= \mathcal{M}^{\mathrm{adj}}(W^{-1}\mathcal{M}(\Delta x)W^{-1}). \end{aligned}$$

The exact expressions for the righthand sides $r_1$, $r_2$, and the positive definite scaling matrix $W$ are not important for our present purposes and are omitted; see Appendix A for details.

The coefficient matrix in (36) is dense, so the cost of solving these equations is $(1/3)(n(n+1)/2+p)^3 = O(n^6)$ operations if we assume that $p = O(n)$. This gives a lower bound for the cost of one iteration of a general-purpose interior-point solver applied to (29). The actual cost is higher since it includes the cost of assembling the matrices $H_{11}$, $H_{12}$, and $H_{22}$.

### 4.2 Dual Formulation

A reformulation based on SDP duality allows us to solve KYP-SDPs more efficiently, at a cost of roughly $O(n^4)$ per iteration. The technique is well known for discrete-time KYP-SDPs with FIR matrices [22, 16, 4, 6], and was applied to general KYP-SDPs in [52].

#### The Reformulated Dual

The assumption that $(A, B)$ is controllable implies that that the mapping $\mathcal{K}$ defined in (31) has full rank, *i.e.*, $\mathcal{K}(P) = 0$ only if $P = 0$. To see this, we can take any stabilizing state feedback matrix $K$, and note that $\mathcal{K}(P) = 0$ implies

$$\begin{bmatrix} I \\ K \end{bmatrix}^T \begin{bmatrix} A^TP + PA & PB \\ B^TP & 0 \end{bmatrix} \begin{bmatrix} I \\ K \end{bmatrix} = (A+BK)^TP + P(A+BK) = 0,$$

and hence $P = 0$. It follows that the nullspace of $\mathcal{K}^{\mathrm{adj}}$ (a linear mapping from $\mathbf{S}^{n+1}$ to $\mathbf{S}^n$) has dimension $n+1$. Hence there exists a mapping $\mathcal{L} : \mathbf{R}^{n+1} \to \mathbf{S}^{n+1}$ that spans the nullspace of $\mathcal{K}^{\mathrm{adj}}$:

$$\begin{aligned} \mathcal{K}^{\mathrm{adj}}(Z) = 0 &\iff Z = \mathcal{L}(u) \text{ for some } u \in \mathbf{R}^{n+1} \\ S = \mathcal{K}(P) \text{ for some } P \in \mathbf{S}^n &\iff \mathcal{L}^{\mathrm{adj}}(S) = 0. \end{aligned}$$

Some practical choices for $\mathcal{L}$ will be discussed later, but first we use this observation to derive an equivalent pair of primal and dual SDPs, with a smaller number of primal and dual variables.

The first equality in the dual SDP (32) is equivalent to saying that $Z = \mathcal{L}(u) - \hat{Z}$ for some $u$, where $\hat{Z}$ is any symmetric matrix that satisfies

$$\mathcal{K}^{\mathrm{adj}}(\hat{Z}) + Q = 0.$$

Substituting in the dual SDP (32), and dropping the constant term $\mathbf{Tr}(N\hat{Z})$ from the objective, we obtain an equivalent problem

$$\begin{array}{ll} \text{maximize} & \mathcal{L}^{\text{adj}}(N)^T u \\ \text{subject to} & \mathcal{L}(u) \succeq \hat{Z} \\ & \mathcal{M}^{\text{adj}}(\mathcal{L}(u)) = q + \mathcal{M}^{\text{adj}}(\hat{Z}) \end{array} \tag{37}$$

with variable $u \in \mathbf{R}^{n+1}$. This SDP has the form (21) with $\mathcal{V} = \mathbf{R}^{n+1}$, $y = u$,

$$\mathcal{A}(u) = -\mathcal{L}(u), \qquad \mathcal{B}(u) = \mathcal{M}^{\text{adj}}(\mathcal{L}(u)),$$

and $c = -\mathcal{L}^{\text{adj}}(N)$, $D = \hat{Z}$, $d = -q - \mathcal{M}^{\text{adj}}(\hat{Z})$.

The dual of problem (37) is

$$\begin{array}{ll} \text{minimize} & (q + \mathcal{M}^{\text{adj}}(\hat{Z}))^T v - \mathbf{Tr}(\hat{Z}S) \\ \text{subject to} & \mathcal{L}^{\text{adj}}(S) - \mathcal{L}^{\text{adj}}(\mathcal{M}(v)) + \mathcal{L}^{\text{adj}}(N) = 0 \\ & S \succeq 0, \end{array} \tag{38}$$

with variables $v \in \mathbf{R}^p$ and $S \in \mathbf{S}^{n+1}$. Not surprisingly, the SDP (38) can be interpreted as a reformulation of the original primal problem (30). The first constraint in (38) is equivalent to

$$S - \mathcal{M}(v) + N = \mathcal{K}(P) \tag{39}$$

for some $P$. Combined with $S \succeq 0$, this is equivalent to $\mathcal{K}(P) + \mathcal{M}(v) \succeq N$. Using (39) we can also express the objective function as

$$\begin{aligned} (q + \mathcal{M}^{\text{adj}}(\hat{Z}))^T v - \mathbf{Tr}(\hat{Z}S) &= q^T v + \mathbf{Tr}((\mathcal{M}(v) - S)\hat{Z}) \\ &= q^T v + \mathbf{Tr}(N\hat{Z}) - \mathbf{Tr}(PK^{\text{adj}}(\hat{Z})) \\ &= q^T v + \mathbf{Tr}(N\hat{Z}) + \mathbf{Tr}(PQ). \end{aligned}$$

Comparing this with (30), we see that the optimal $v$ in (38) is equal the optimal $x$ in (30). The relation (39) also allows us to recover the optimal $P$ for (30) from the optimal solution $(v, S)$ of (38).

In summary, the pair of primal and dual SDPs (37) and (38) is equivalent to the original SDPs (30) and (32); the optimal solutions for one pair of SDPs are easily obtained from the solutions of the other pair.

### Newton Equations for Reformulated Dual

A primal-dual method applied to (37) generates iterates $u$, $v$, $S$. At each iteration a set of linear equations of the form (26)–(27) is solved, which in this case reduce to

$$\mathcal{L}^{\text{adj}}(W^{-1}\mathcal{L}(\Delta u)W^{-1}) + \mathcal{L}^{\text{adj}}(\mathcal{M}(\Delta v)) = R \tag{40}$$

$$\mathcal{M}^{\text{adj}}(\mathcal{L}(\Delta v)) = r \tag{41}$$

with variables $\Delta u \in \mathbf{R}^{n+1}$, $\Delta v \in \mathbf{R}^p$. (Again, we omit the expressions for $W$, $R$, $r$. In particular, note that $W$ is not the same matrix as in §4.1.) In matrix form,

$$\begin{bmatrix} H & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} R \\ r \end{bmatrix}, \tag{42}$$

where $H$ and $G$ are defined by the identities

$$H\Delta u = \mathcal{L}^{\mathrm{adj}}(W^{-1}\mathcal{L}(\Delta u)W^{-1}), \qquad G\Delta v = \mathcal{L}^{\mathrm{adj}}(\mathcal{M}(\Delta v)).$$

The number of variables in (42) is $p + n + 1$.

**Computational Cost**

We now estimate the cost of assembling the coefficient matrix in (42), for a specific choice of $\mathcal{L}$. To simplify the notation, we assume that the Lyapunov operator $AX + XA^T$ is invertible. This assumption can be made without loss of generality: Since $(A, B)$ is controllable by assumption, there exists a state feedback matrix $K$ such that $A + BK$ is stable (or, more generally, $\lambda_i(A+BK)+\lambda_j(A+BK)^* \neq 0$, for $i, j = 1, \ldots, n$). By applying a congruence to both sides of the LMI constraint in (29) and noting that

$$\begin{bmatrix} I & K^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A^TP + PA & PB \\ B^TP & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ K & I \end{bmatrix} = \begin{bmatrix} (A+BK)^TP + P(A+BK) & PB \\ B^TP & 0 \end{bmatrix},$$

we can transform the SDP (29) to an equivalent KYP-SDP

$$\begin{array}{ll} \text{minimize} & q^Tx + \mathbf{Tr}(QP) \\ \text{subject to} & \begin{bmatrix} (A+BK)^TP + P(A+BK) & PB \\ B^TP & 0 \end{bmatrix} + \sum_{i=1}^p x_i\tilde{M}_i \succeq \tilde{N}, \end{array}$$

where

$$\tilde{M}_i = \begin{bmatrix} I & K^T \\ 0 & I \end{bmatrix} M_i \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}, \qquad \tilde{N} = \begin{bmatrix} I & K^T \\ 0 & I \end{bmatrix} N \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}.$$

We will therefore assume that the matrix $A$ in (29) is stable.

It is then easily verified that $\mathcal{K}^{\mathrm{adj}}(Z) = 0$ if and only if $Z = \mathcal{L}(u)$ for some $u$, with $\mathcal{L}$ defined as

$$\mathcal{L}(u) = \sum_{i=1}^{n+1} u_iF_i,$$

where

$$F_i = \begin{bmatrix} X_i & e_i \\ e_i^T & 0 \end{bmatrix}, \quad i = 1, \ldots, n, \qquad F_{n+1} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}. \tag{43}$$

and $X_i$, $i = 1, \ldots, n$, are the solutions of the Lyapunov equations

$$AX_i + X_iA^T + Be_i^T + e_iB^T = 0. \tag{44}$$

With this choice of $\mathcal{L}$, the coefficient matrices $H$ and $G$ in (42) can be expressed as

$$H_{ij} = \mathbf{Tr}(F_i W^{-1} F_j W^{-1})), \quad i, j = 1, \ldots, n+1, \tag{45}$$

$$G_{ij} = \mathbf{Tr}(F_i M_j), \quad i = 1, \ldots, n+1, \quad j = 1, \ldots, p. \tag{46}$$

To estimate the cost of this approach we assume that $p = O(n)$. The method requires a significant amount of preprocessing. In particular we have to compute the solutions $X_i$ of $n+1$ Lyapunov equations, which has a total cost of $O(n^4)$. The matrix $G$ does not change during the algorithm so it can be pre-computed, at a cost of order $pn^3$ if the matrices $M_i$ and $X_j$ are dense (*i.e.*, $O(n^4)$ if we assume $p = O(n)$). In practice, as we have seen in §2, the matrices $M_i$ are often sparse or low-rank, so the cost of computing $G$ is usually much lower than $O(n^4)$.

At each iteration, we have to construct $H$ and solve the equations (42). The cost of constructing $H$ is $O(n^4)$. The cost of solving the equations is $O(n^3)$ if we assume $p = O(n)$. The total cost is therefore $O(n^4)$, and is dominated by the cost of pre-computing the basis matrices $X_i$, and the cost of forming $H$ at each iteration.

## 5 Special-Purpose Implementation

We now turn to the question of exploiting additional problem structure in a special-purpose implementation. As should be clear from the previous section, the key to a fast implementation is to solve the linear equations that arise in each iteration fast. This can be done for either the primal or the dual formulation described in §4. We will see that these two approaches lead to methods that are almost identical, and have the same complexity.

### 5.1 Reduced Newton Equations

In §4 we noted a large difference in complexity between solving the original KYP-SDP (29) and solving the reformulated dual problem (37). The difference is due to the different dimension and structure of the Newton equations in each iteration, and the way in which special-purpose codes handle those equations. In a custom implementation, the distinction between the two formulations disappears: the equations (33)–(35) that arise when solving the primal formulation can be solved as efficiently as the equations (40)–(41) that arise when solving the dual formulation.

#### Solving Newton Equations via Dual Elimination

To show this, we describe an alternative method for solving (33)–(35). As in §4.2, let $\mathcal{L} : \mathbf{R}^{n+1} \to \mathbf{S}^{n+1}$ be a linear mapping that spans the nullspace of

$\mathcal{K}^{\rm adj}$. Let $Z_0$ be any symmetric matrix that satisfies $\mathcal{K}^{\rm adj}(Z_0) + R_2 = 0$. The equation (34) is equivalent to saying that

$$\Delta Z = \mathcal{L}(\Delta u) - Z_0$$

for some $\Delta u \in \mathbf{R}^{n+1}$. Substituting this expression in (33) and (35), we obtain

$$\begin{aligned} W\mathcal{L}(\Delta u)W + \mathcal{K}(\Delta P) + \mathcal{M}(\Delta x) &= R_1 + WZ_0W \\ \mathcal{M}^{\rm adj}(\mathcal{L}(\Delta u)) &= r + \mathcal{M}^{\rm adj}(Z_0). \end{aligned}$$

Next we eliminate the variable $\Delta P$, by applying $\mathcal{L}^{\rm adj}$ to both sides of the first equation, and using the fact that $\mathcal{L}^{\rm adj}(\mathcal{K}(\Delta P)) = 0$ for all $\Delta P$:

$$\mathcal{L}^{\rm adj}(W\mathcal{L}(\Delta u)W) + \mathcal{L}^{\rm adj}(\mathcal{M}(\Delta x)) = \mathcal{L}^{\rm adj}(R_1 + WZ_0W) \tag{47}$$

$$\mathcal{M}^{\rm adj}(\mathcal{L}(\Delta u)) = r + \mathcal{M}^{\rm adj}(Z_0). \tag{48}$$

This is a set of $n + p + 1$ linear equations in $n + p + 1$ variables $\Delta u$, $\Delta x$. In matrix form,

$$\begin{bmatrix} H & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta x \end{bmatrix} = \begin{bmatrix} \mathcal{L}^{\rm adj}(R_1 + WZ_0W) \\ r + \mathcal{M}^{\rm adj}(Z_0) \end{bmatrix}, \tag{49}$$

where $H$ and $G$ are defined by the identities

$$H\Delta u = \mathcal{L}^{\rm adj}(W\mathcal{L}(\Delta u)W), \qquad G\Delta x = \mathcal{L}^{\rm adj}(\mathcal{M}(\Delta x)).$$

Since $\mathcal{L}$ has full rank, the matrix $H$ is nonsingular, so the equations (49) can be solved by first solving

$$G^TH^{-1}G\Delta x = G^TH^{-1}\mathcal{L}^{\rm adj}(R_1 + WZ_0W) - r - \mathcal{M}^{\rm adj}(Z_0)$$

to obtain $\Delta x$, and then computing $\Delta u$ from

$$H\Delta u = \mathcal{L}^{\rm adj}(R_1 + WZ_0W) - G\Delta x.$$

After solving (49), we can compute $\Delta Z$ as $\Delta Z = \mathcal{L}(\Delta u)$. Given $\Delta Z$ and $\Delta x$, we find $\Delta P$ by solving

$$\mathcal{K}(\Delta P) = R_1 - W\Delta ZW - \mathcal{M}(\Delta x),$$

which is an overdetermined, but solvable set of linear equations.

We will refer to (49) as the *reduced Newton equations*.

**Computational Cost**

We now estimate the complexity of solving the reduced Newton equations. Note that (47)–(48) have exactly the same form as (40)–(41), with different values of $W$ and the righthand sides. In particular, our discussion of the complexity of solving (40)–(41) also applies here.

We work out the details assuming the Lyapunov operator $AX + XA^T$ is invertible. If this is not the case, the equations (33)–(35) can be transformed into an equivalent set

$$\begin{aligned} TWT^T(T^{-T}\Delta Z T^{-1})TWT^T + T\mathcal{K}(\Delta P)T^T + T\mathcal{M}(\Delta x)T^T &= TR_1T^T \\ \mathcal{K}^{\mathrm{adj}}(T^TT^{-T}\Delta Z T^{-1}T) &= R_2 \\ \mathcal{M}^{\mathrm{adj}}(T^TT^{-T}\Delta Z T^{-1}T) &= r, \end{aligned}$$

where

$$T = \begin{bmatrix} I & K^T \\ 0 & I \end{bmatrix}, \qquad T^{-1} = \begin{bmatrix} I & -K^T \\ 0 & I \end{bmatrix}$$

and $K$ is a stabilizing state feedback matrix. Replacing $\Delta Z$ with a new variable

$$\Delta S = T^{-T}\Delta Z T^{-1},$$

gives

$$\tilde{W}\Delta S\tilde{W} + \begin{bmatrix} (A+BK)^T\Delta P + \Delta P(A+BK) & \Delta PB \\ B^T\Delta P & 0 \end{bmatrix} + \sum_{i=1}^{p} \Delta x_i \tilde{M}_i = \tilde{R}_1$$

$$\begin{bmatrix} A+BK \\ B \end{bmatrix}^T \Delta S \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix}^T \Delta S \begin{bmatrix} A+BK \\ B \end{bmatrix} = R_2$$

$$\mathbf{Tr}(\tilde{M}_i\Delta S) = r_i, \quad i = 1, \ldots, p,$$

where $\tilde{W} = TWT^T$, $\tilde{M}_i = TM_iT^T$, $\tilde{R}_1 = TR_1T^T$. These equations have the same structure as the original equations (33)–(35), with $A$ replaced by $A + BK$.

If we assume that $AX + XA^T$ is invertible, we can choose $\mathcal{L}$ as in §4.2: $\mathcal{L}(u) = \sum_{i=1}^{n+1} u_iF_i$ with the matrices $F_i$ defined as in (43). If the matrices $X_i$ are pre-computed (at a total cost of $O(n^4)$), then the cost of constructing the coefficient matrix $H$ in (49) is $O(n^4)$. The cost of computing $G$ is $O(pn^3)$ if we assume the matrices $X_i$ are known, and we do not exploit any particular structure in $M_j$. The cost of solving the equations (49), given $H$ and $G$, is $O(n^3)$ if we assume $p = O(n)$.

**Comparison with Dual Method**

The above analysis demonstrates that a custom implementation of a primal-dual method for solving the original KYP-SDP (29) can be as efficient as a primal-dual method applied to the reformulated dual (37). Both methods are based on eliminating dual variables, either in the original dual SDP, or in the Newton equations. In fact, if we use the same mapping $\mathcal{L}$ in both methods, the reduced linear equations are identical. However, a custom implementation offers three important advantages over a general-purpose primal-dual method applied to the reformulated dual.

- In a custom implementation we can avoid the need to compute and store the basis matrices $F_i$ (or $X_i$), which are required in the dual formulation (see §5.2 for details). These matrices $X_i$ are the solution of $n$ Lyapunov equations of order $n$. For large $n$, they are expensive to compute and store.
- Additional problem structure can be exploited in a custom implementation. Two methods that achieve an $O(n^3)$ complexity per iteration are described in §5.2.
- In a custom implementation, we can make a different choice for the mapping $\mathcal{L}$, which is used to eliminate dual variables, in each iteration.
  For example, in §4.2 we pointed out that state feedback transformations preserve the KYP structure in the SDP, while in §5.1 we made a similar observation about the Newton equations. Of course, these two viewpoints are just different interpretations of the same property. We can first use state feedback to transform the SDP and then derive the Newton equations, or we can write down Newton equations for the original SDP and then apply a state feedback transformation. Both transformations result in the same equations. However the second viewpoint opens the possibility of selecting a different state-feedback matrix $K$ in each iteration, in order to improve the numerical stability of the elimination step.

### 5.2 Fast Construction of Reduced Newton Equations

We now examine two methods that allow us to construct the matrices $H$ and $G$ in (49) fast, in roughly $O(n^3)$ operations.

#### Diagonalizable $A$

Suppose $A$ is stable (or equivalently, a state feedback transformation has been applied as described in §5.1, to obtain equivalent equations with a stable $A$). We make the same choice of $\mathcal{L}$ as in §5.1, *i.e.*, $\mathcal{L}(u) = \sum_{i=1}^{n+1} u_i F_i$, with $F_i$ defined in (43).

In appendix B we derive the following expression for the matrix $H$ in (49):

$$\begin{aligned} H = &\begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix} + 2 \begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix} \begin{bmatrix} H_2 & 0 \end{bmatrix} + 2 \begin{bmatrix} H_2^T \\ 0 \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \end{bmatrix} \\ &+ 2W_{22}W + 2 \begin{bmatrix} W_{12} \\ W_{22} \end{bmatrix} \begin{bmatrix} W_{21} & W_{22} \end{bmatrix} \end{aligned} \tag{50}$$

where

$$(H_1)_{ij} = \mathbf{Tr}(X_i W_{11} X_j W_{11}), \qquad H_2 = \begin{bmatrix} X_1 W_{12} & X_2 W_{12} & \cdots & X_n W_{12} \end{bmatrix} \tag{51}$$

and

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

with $W_{11} \in \mathbf{S}^{n\times n}$. Similarly,

$$G = 2\begin{bmatrix} G_1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} M_{1,12} & M_{2,12} & \cdots & M_{p,12} \\ M_{1,22} & M_{2,22} & \cdots & M_{p,22} \end{bmatrix}, \tag{52}$$

where

$$G_1 = \begin{bmatrix} Y_1 B & Y_2 B & \cdots & Y_p B \end{bmatrix},$$

$Y_j$ is the solution of

$$AY_j + Y_j A^T + M_{j,11} = 0,$$

and $M_{j,11} \in \mathbf{S}^n$, $M_{j,12} \in \mathbf{R}^n$ are the $1,1$- and $1,2$-blocks of $M_j$. Formulas (50) and (52) show that the key to constructing $H$ and $G$ fast (*i.e.*, faster than in $O(n^4)$ and $O(pn^3)$ operations, respectively), is to compute the matrices $H_1$, $H_2$, and $G_1$ fast.

A simple approach is based on the eigenvalue decomposition of $A$. Our assumption that $(A, B)$ is controllable implies that is possible to find a linear state feedback matrix $K$ so that $A + BK$ is stable and diagonalizable [35]. As mentioned in §5.1, we can transform the Newton equations into an equivalent set of equations in which the matrix $A$ is replaced by $A+BK$. We can therefore assume without loss of generality that $A$ is diagonalizable.

Let $A = V\,\mathbf{diag}(\lambda)V^{-1}$ be the eigenvalue decomposition of $A$, with $V \in \mathbf{C}^{n\times n}$ and $\lambda \in \mathbf{C}^n$. It can be shown that the matrices $H_1$ and $H_2$ defined in (51) can be expressed as

$$H_1 = 2\,\mathrm{Re}\left(\left(V^{-T}((\widetilde{\Sigma}\widetilde{W}_{11}) \circ (\widetilde{\Sigma}\widetilde{W}_{11})^T) + V^{-*}(\widetilde{W}_{11} \circ (\widetilde{\Sigma}\widetilde{W}_{11}\widetilde{\Sigma}^*)^T)\right) V^{-1}\right) \tag{53}$$

$$H_2 = -V(\widetilde{\Sigma}^*\,\mathbf{diag}(\widetilde{W}_{12}))\bar{V}^{-1} - V\,\mathbf{diag}(\widetilde{\Sigma}\widetilde{W}_{12})V^{-1} \tag{54}$$

where $\circ$ denotes Hadamard product, $\Sigma \in \mathbf{C}^{n\times n}$ is defined as

$$\Sigma_{ij} = \frac{1}{\lambda_i + \lambda_j^*}, \qquad i, j = 1, \ldots, n,$$

$\widetilde{\Sigma} = \Sigma\,\mathbf{diag}(V^{-1}B)^*$, $\widetilde{W}_{11} = V^*W_{11}V$, $\widetilde{W}_{12} = V^*W_{12}$, and $\bar{V}$ is the complex conjugate of $V$. The above formulas for $H_1$ and $H_2$ can be evaluated in $O(n^3)$ operations, and do not require pre-computing the basis matrices $X_i$. We refer to appendix C for a proof of the expressions (53) and (54).

There is a similar expression for $G_1$:

$$G_1 = V\begin{bmatrix} (\widetilde{M}_1 \circ \Sigma)V^*B & (\widetilde{M}_2 \circ \Sigma)V^*B & \cdots & (\widetilde{M}_p \circ \Sigma)V^*B \end{bmatrix}.$$

where $\widetilde{M}_j = V^{-1}M_{j,11}V^{-*}$. The cost of computing $\widetilde{M}_j$ can be reduced by exploiting low-rank structure in $M_{j,11}$. Given the matrices $\widetilde{M}_j$, $G_1$ can be computed in $O(n^2p)$ operations.

### *A* in Companion Form

In this section we present an $O(n^3)$ method for solving the Newton equations when $A$ is an $n \times n$ 'shift matrix' and $B = e_n$:

$$A = \begin{bmatrix} 0\,1\,0\cdots 0 \\ 0\,0\,1\cdots 0 \\ \vdots\,\vdots\,\vdots\,\ddots\,\vdots \\ 0\,0\,0\cdots 1 \\ 0\,0\,0\cdots 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{55}$$

KYP-SDPs of this form arise in a wide variety of LMI problems in robust control [27]. The method is also useful for handling matrices $A$ in companion form: in order to solve the Newton equations for a KYP-SDP with

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_1 & -a_2 & -a_3 & \cdots & -a_n \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

we can first apply a state feedback transformation with

$$K = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix},$$

as explained in §5.1, and then solve an equivalent set of equations in which $A$ is replaced with the shift matrix $A + BK$.

With $A$ and $B$ defined as in (55), the equations (33)–(35) reduce to

$$W\Delta ZW + \begin{bmatrix} 0 & 0 \\ \Delta P & 0 \end{bmatrix} + \begin{bmatrix} 0 & \Delta P \\ 0 & 0 \end{bmatrix} + \sum_{i=1}^{p} \Delta x_i M_i = R_1 \tag{56}$$

$$\begin{bmatrix} 0 & I \end{bmatrix} \Delta Z \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} I & 0 \end{bmatrix} \Delta Z \begin{bmatrix} 0 \\ I \end{bmatrix} = R_2 \tag{57}$$

$$\mathbf{Tr}(M_i \Delta Z) = r_i, \quad i = 1, \ldots, p. \tag{58}$$

We can follow the method of §5.1, with $\mathcal{L} : \mathbf{R}^{n+1} \to \mathbf{S}^{n+1}$ defined as

$$\mathcal{L}(u) = \begin{bmatrix} u_1 & 0 & u_2 & 0 & u_3 & \cdots & 0 & u_{k+1} \\ 0 & -u_2 & 0 & -u_3 & 0 & \cdots & -u_{k+1} & 0 \\ u_2 & 0 & u_3 & 0 & u_4 & \cdots & 0 & u_{k+2} \\ 0 & -u_3 & 0 & -u_4 & 0 & \cdots & -u_{k+2} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ u_k & 0 & u_{k+1} & 0 & u_{k+2} & \cdots & 0 & u_{2k} \\ 0 & -u_{k+1} & 0 & -u_{k+2} & 0 & \cdots & -u_{2k} & 0 \\ u_{k+1} & 0 & u_{k+2} & 0 & u_{k+1} & \cdots & 0 & u_{2k+1} \end{bmatrix}, \qquad n = 2k$$

$$\mathcal{L}(u) = \begin{bmatrix} u_1 & 0 & u_2 & 0 & u_3 & \cdots & u_{k+1} & 0 \\ 0 & -u_2 & 0 & -u_3 & 0 & \cdots & 0 & -u_{k+2} \\ u_2 & 0 & u_3 & 0 & u_4 & \cdots & u_{k+2} & 0 \\ 0 & -u_3 & 0 & u_4 & 0 & \cdots & 0 & -u_{k+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & -u_{k+1} & 0 & -u_{k+2} & 0 & \cdots & 0 & -u_{2k+1} \\ u_{k+1} & 0 & u_{k+2} & 0 & u_{k+3} & \cdots & u_{2k+1} & 0 \\ 0 & -u_{k+2} & 0 & -u_{k+3} & 0 & \cdots & 0 & -u_{2k+2} \end{bmatrix}, \qquad n = 2k+1.$$

In other words, the even anti-diagonals of $\mathcal{L}(u)$ are zero. The elements on the odd anti-diagonals have equal absolute values and alternating signs. The nonzero elements in the first row and column are given by $u$.

To obtain efficient formulas for the matrix $H$ in (49), we represent $\mathcal{L}$ as $\mathcal{L}(u) = \sum_{i=1}^{n+1} u_i F_i$, where

$$(F_i)_{jk} = \begin{cases} (-1)^{j+1} & j+k = 2i \\ 0 & \text{otherwise.} \end{cases}$$

We also factor $W$ as $W = \sum_{k=1}^{n+1} v_k v_k^T$ (for example, using the Cholesky factorization or the eigenvalue decomposition). The $i,j$-element of $H$ is

$$H_{ij} = \mathbf{Tr}(F_i W F_j W) = \sum_{k=1}^{n+1} \sum_{l=1}^{n+1} (v_l^T F_i v_k)(v_k^T F_j v_l).$$

Next we note that for $v, w \in \mathbf{R}^{n+1}$,

$$\begin{aligned} v^T F_i w &= \sum_{j+k=2i} (-1)^{j+1} v_j w_k \\ &= \sum_{k=\max\{1,2i-n-1\}}^{\min\{n+1,2i-1\}} (-1)^{k+1} w_k v_{2i-k} \\ &= (v * (Dw))_{2i-1}, \end{aligned}$$

where $D = \mathbf{diag}(1, -1, 1, \ldots)$, and $v * (Dw)$ denotes the convolution of the vectors $v$ and $Dw$. Therefore,

$$(v_l^T F_1 v_k, v_l^T F_2 v_k, \ldots, v_l^T F_{n+1} v_k) = E(v_l * (Dv_k))$$

where

$$E = \begin{bmatrix} 1\,0\,0\,0\cdots 0\,0 \\ 0\,0\,1\,0\cdots 0\,0 \\ \vdots\;\vdots\;\vdots\;\vdots\quad\;\vdots\;\vdots \\ 0\,0\,0\,0\cdots 0\,1 \end{bmatrix} \in \mathbf{R}^{(n+1)\times(2n+1)}.$$

Using this notation we can express $H$ as

$$H = E\left(\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}(v_l * (Dv_k))(v_k * (Dv_l))^T\right)E^T.$$

This expression can be evaluated in $O(n^3)$ operations using the discrete Fourier transform (DFT). Let $W_{\mathrm{DFT}} \in \mathbf{C}^{(2n+1)\times(n+1)}$ be the first $n+1$ columns of the DFT matrix of length $2n+1$, *i.e.*, $W_{\mathrm{DFT}}v$ is the zero padded $(2n+1)$-point DFT of a vector $v \in \mathbf{R}^{n+1}$. Let

$$V_k = W_{\mathrm{DFT}}v_k, \qquad \tilde{V}_k = W_{\mathrm{DFT}}Dv_k, \qquad k = 1,\ldots,n+1,$$

be the DFTs of the vectors $v_k$ and $Dv_k$. Then

$$\begin{aligned} H &= \frac{1}{(2n+1)^2}EW_{\mathrm{DFT}}^*\left(\sum_{k=1}^{n+1}\sum_{l=1}^{n+1}(V_l \circ \tilde{V}_k)(V_k \circ \tilde{V}_l)^*\right)W_{\mathrm{DFT}}E^T \\ &= \frac{1}{(2n+1)^2}EW_{\mathrm{DFT}}^*\left(\left(\sum_{l=1}^{n+1}V_l\tilde{V}_l^*\right)\circ\left(\sum_{k=1}^{n+1}\tilde{V}_kV_k^*\right)\right)W_{\mathrm{DFT}}E^T. \end{aligned}$$

The matrix in the middle is the Hadamard product of the $(2n+1)\times(2n+1)$ matrix $\sum_k V_k\tilde{V}_k^*$ with its conjugate transpose, so the cost of evaluating this expression is $O(n^3)$.

The matrix $G$ in (49) has elements

$$G_{ij} = \mathbf{Tr}(F_iM_j), \quad i = 1,\ldots,n+1, \quad j = 1,\ldots,p,$$

and is easily computed in $O(n^2p)$ operations, since only $O(n)$ elements of $F_i$ are nonzero. For sparse or low-rank $M_j$ the cost is even lower.

## 6 Numerical Examples

In Table 2 we compare four algorithms, applied to randomly generated KYP-SDPs of the form (29), with dimensions $n = 25, 50, 100, 200$, and $p = n$. Each problem was constructed so it is strictly primal and dual feasible. (However the algorithms were started at infeasible starting points.) The execution times listed are the CPU times in seconds on a 2GHz Pentium IV PC with 1GB of memory. All times are averages over five randomly generated instances.

**Table 2.** Computational results for the general-purpose SDP solvers SeDuMi and SDPT3 applied to KYP-SDPs with dimensions $n = p = 25, \ldots, 200$, applied to the original problem ('Primal'), and to the reformulated dual SDP ('Dual'). $T_{\mathrm{p}}$ is the time in seconds required for preprocessing, and consists mainly of the cost of computing an explicit basis for the solution set of the dual equality constraints. $T_{\mathrm{s}}$ is the solution time per iteration, excluding preprocessing time.

| | SeDuMi (primal) | | SDPT3 (primal) | | SeDuMi (dual) | | | SDPT3 (dual) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | #itrs | $T_{\mathrm{s}}$ | #itrs | $T_{\mathrm{s}}$ | $T_{\mathrm{p}}$ | #itrs | $T_{\mathrm{s}}$ | $T_{\mathrm{p}}$ | #itrs | $T_{\mathrm{s}}$ |
| 25 | 10 | 0.3 | 8 | 0.8 | 0.1 | 12 | 0.04 | 0.1 | 9 | 0.06 |
| 50 | 11 | 8.1 | 9 | 4.9 | 1.1 | 11 | 0.3 | 1.1 | 9 | 0.26 |
| 100 | 11 | 307.1 | 8 | 107.2 | 21.4 | 14 | 3.3 | 21.4 | 10 | 1.4 |
| 200 | | | | | 390.7 | 12 | 30.9 | 390.7 | 10 | 15.3 |

The first method, SeDuMi (primal), solves the SDP (29) using the general-purpose solver SeDuMi (version 1.05R5) [44]. The second method, SDPT3 (primal), solves the same problem using the general-purpose solver SDPT3 (version 3.02) [47]. Both solvers were called via the YALMIP interface [36]. We skip the last problem ($n = 200$), due to excessive computation time and memory requirements. The numbers $T_{\mathrm{s}}$ are the CPU times needed to solve each problem, divided by the number of iterations.

The other methods, SeDuMi (dual) and SDPT3 (dual), solve the reformulated dual problem (37), for the choice of basis matrices described in §5.1. In addition to the number of iterations and the time per iteration $T_{\mathrm{s}}$, we also give $T_{\mathrm{p}}$, the preprocessing time required to compute the parameters in the reformulated dual problem (37). This preprocessing step is dominated by the cost of of solving the $n + 1$ Lyapunov equations (44).

The results confirm that the cost per iteration of a general-purpose method applied to the primal SDP (29) grows much more rapidly than the same method applied to the reformulated dual problem (37).

Table 3 shows the results for a second experiment with randomly generated KYP-SDPs of dimensions $n = 100, \ldots, 500$, and $p = 50$. Again, all values are averages over five problem instances. The data in the column KYP-IPM are for a customized Matlab implementation of the primal-dual interior-point method of Tütüncü, Toh, and Todd [46, 47], applied to the dual problem, and using the expressions (53) and (54) to compute the coefficient matrix of the reduced Newton equations. The preprocessing time for this method includes the eigenvalue decomposition of $A$ and the computation of the matrix $G$ in the reduced system (42). The table shows that the preprocessing time and execution time per iteration grow almost exactly as $n^3$.

For the same set of problems, we also show the results for SeDuMi applied to the reformulated dual problem. To speed up the calculation in SeDuMi, we first transform the dual problem (37), by diagonalizing $A$. This corresponds to a simple change of variables, replacing $Z$ with $V^{-1}ZV^{-*}$ and $\tilde{z}$ with $V^{-1}\tilde{z}$.

**Table 3.** Results for KYP-SDPs of dimension $n = 100,\ldots, n = 500$, and $p = 50$. The first method is a customized Matlab implementation of a primal-dual method as described in §5, using the formulas (53) and (54). The second method is SeDuMi applied to the reformulated dual SDP (37), after first diagonalizing $A$. The third method solves the same reformulated dual SDP using SDPT3, without the diagonalization of $A$ (except in the preprocessing).

| | KYP-IPM | | | SeDuMi (dual) | | | SDPT3 (dual) | | |
|---|---|---|---|---|---|---|---|---|---|
| $n$ | $T_{\rm p}$ | #itrs | $T_{\rm s}$ | $T_{\rm p}$ | #itrs | $T_{\rm s}$ | $T_{\rm p}$ | #itrs | $T_s$ |
| 100 | 1.0 | 9 | 0.6 | 0.5 | 12 | 1.5 | 3.6 | 12 | 1.3 |
| 200 | 8.3 | 9 | 4.7 | 3.5 | 13 | 24.4 | 44.4 | 13 | 13.8 |
| 300 | 28.1 | 10 | 16.7 | 11.7 | 12 | 155.3 | 194.2 | 14 | 77.7 |
| 400 | 62.3 | 10 | 36.2 | 26.7 | 12 | 403.7 | | | |
| 500 | 122.0 | 10 | 70.3 | 51.9 | 12 | 1068.4 | | | |

We then eliminate the (1,1)-block in the dual variable as in the SeDuMi (dual) method of Table 2, which gives a problem of form (37), with complex data and variables. Since $A$ is diagonal, the basis matrices $X_i$ are quite sparse and easier to compute (at a cost of $O(n^3)$ total). Despite the resulting savings, it is clear from the table that the execution time per iteration grows roughly as $n^4$.

The third column (SDTP3 (dual)) gives the results for SDPT3 applied to the reformulated dual problem. Since the version of SDPT3 we used does not accept complex data, we only used diagonalization of $A$ in the preprocessing step, to accelerate the solution of the Lyapunov equations (44). Results are reported for the first three problems only, due to insufficient memory. As for SeDuMi, the results show an $O(n^4)$-growth for the solution time per iteration.

## 7 Extensions

In this section we discuss some extensions of the techniques of §4 and §5 to the general problem (1).

### 7.1 Multiple Constraints

Consider a KYP-SDP with multiple constraints,

$$\begin{array}{ll} \text{minimize} & q^T x + \sum_{k=1}^{L} (Q_k P_k) \\ \text{subject to} & \mathcal{K}_k(P_k) + \mathcal{M}_k(x) \succeq N_k, \quad k = 1, \ldots, L, \end{array}$$

where $\mathcal{K}_k : \mathbf{S}^{n_k} \to \mathbf{S}^{n_k+1}$ and $\mathcal{M}_k : \mathbf{R}^p \to \mathbf{S}^{n_k+1}$ are defined as

$$\mathcal{K}_k(P_k) = \begin{bmatrix} A_k^T P_k + P_k A_k & P_k B_k \\ B_k^T P_k & 0 \end{bmatrix}, \qquad \mathcal{M}_k(x) = \sum_{i=1}^{p} x_i M_{ki}.$$

We assume $(A_k, B_k)$ is controllable for $k = 1, \ldots, L$. The Newton equations that need to be solved at each iteration take the form

$$\begin{aligned} W_k \Delta Z_k W_k + \mathcal{K}_k(\Delta P_k) + \mathcal{M}_k(\Delta x) &= R_{\mathrm{pri},k}, \quad k = 1, \ldots, L \\ \mathcal{K}_k^{\mathrm{adj}}(\Delta Z_k) &= R_{\mathrm{du},k}, \quad k = 1, \ldots, L \\ \sum_{k=1}^{L} \mathcal{M}_k^{\mathrm{adj}}(\Delta Z_k) &= r, \end{aligned}$$

with variables $\Delta P_k \in \mathbf{S}^{n_k}$, $\Delta x \in \mathbf{R}^p$, $\Delta Z_k \in \mathbf{S}^{n_k+1}$. The values of the positive definite matrix $W_k$ and the righthand sides change at each iteration. As in the single-constraint case, we solve the Newton equations by eliminating some of the dual variables, and expressing $\Delta Z_k$ as

$$\Delta Z_k = \mathcal{L}_k(\Delta u_k) - \hat{Z}_k,$$

where $\mathcal{L}_k : \mathbf{R}^{n_k+1} \to \mathbf{S}^{n_k+1}$ parametrizes the nullspace of $\mathcal{K}_k^{\mathrm{adj}}$, and $\hat{Z}_k$ satisfies

$$\mathcal{K}_k^{\mathrm{adj}}(\hat{Z}_k) + R_{\mathrm{du},k} = 0.$$

We then apply $\mathcal{L}_k^{\mathrm{adj}}$ to both sides of the first group of equations and obtain

$$\mathcal{L}_k^{\mathrm{adj}}(W \mathcal{L}_k(\Delta u_k) W) + \mathcal{L}_k^{\mathrm{adj}}(\mathcal{M}_k(\Delta x)) = \mathcal{L}_k^{\mathrm{adj}}(R_{\mathrm{pri},k} + W_k \hat{Z}_k W_k),$$

for $k = 1, \ldots, L$, and

$$\sum_{k=1}^{L} \mathcal{M}_k^{\mathrm{adj}}(\mathcal{L}_k(\Delta u_k)) = r + \sum_{k=1}^{L} \mathcal{M}_k^{\mathrm{adj}}(\hat{Z}_k).$$

In matrix form,

$$\begin{bmatrix} H_1 & 0 & \cdots & 0 & G_1 \\ 0 & H_2 & \cdots & 0 & G_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & H_L & G_L \\ G_1^T & G_2^T & \cdots & G_L^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_L \\ \Delta x \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1^{\mathrm{adj}}(R_{\mathrm{pri},1} + W_1 \hat{Z}_1 W_1) \\ \mathcal{L}_2^{\mathrm{adj}}(R_{\mathrm{pri},2} + W_2 \hat{Z}_2 W_2) \\ \vdots \\ \mathcal{L}_L^{\mathrm{adj}}(R_{\mathrm{pri},L} + W_L \hat{Z}_L W_L) \\ r + \sum_{k=1}^{L} \mathcal{M}_k^{\mathrm{adj}}(\hat{Z}_k) \end{bmatrix}. \tag{59}$$

To solve these equations we first solve

$$\sum_{k=1}^{L} G_k^T H_k^{-1} G_k \Delta x = -r - \sum_{k=1}^{L} \left( \mathcal{M}_k^{\mathrm{adj}}(\hat{Z}_k) - G_k^T H_k^{-1} \mathcal{L}_k^{\mathrm{adj}}(R_k + W_k \hat{Z}_k W_k) \right) \tag{60}$$

for $\Delta x$, and then solve

$$H_k \Delta u_k = \mathcal{L}_k^{\mathrm{adj}}(R_{\mathrm{pri},k} + W_k \hat{Z}_k W_k) - G_k \Delta x, \quad k = 1, \ldots, L,$$

to determine $\Delta u_k$. As in the single-constraint case, the cost of this method is dominated by the cost of forming the coefficient matrices $H_k$ and $G_k$, the coefficient matrix of (60), and the cost of solving this system. The matrices $G_k$ can be pre-computed. Assuming $n_k = O(n)$ for $k = 1, \ldots, L$, the cost of forming $H_k$ is $O(n^4)$, or $O(n^3)$ if we use one of the methods of §5. Assuming $p = O(n)$, the cost of forming and solving the equations (60) is $O(Ln^3)$. Overall, the cost increases linearly with $L$.

### 7.2 Multivariable Systems

The extension to systems with multiple inputs ($m_k > 1$) is also quite straightforward, although the formulas get more involved. The Newton equations include constraints

$$\mathcal{K}_k^{\mathrm{adj}}(\Delta Z_k) = \begin{bmatrix} A_k \\ B_k \end{bmatrix}^T \Delta Z_k \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix}^T \Delta Z_k \begin{bmatrix} A \\ B \end{bmatrix} = R_{\mathrm{pri},k}, \qquad k = 1, \ldots, L.$$

By representing $\Delta Z_k$ as

$$\Delta Z_k = \mathcal{L}_k(\Delta u_k) - \hat{Z}_k$$

where $\mathcal{L} : \mathbf{R}^{n_k m_k + m_k(m_k+1)/2} \to \mathbf{S}^{m_k+n_k}$, we can obtain reduced Newton equations of the form (59). If $m_k \ll n_k$ this offers a practical and efficient alternative to standard general-purpose methods.

## 8 Conclusion

We have described techniques for exploiting problem structure in interior-point methods for KYP-SDPs, a class of large-scale SDPs that are common in control applications. The method is very effective if the SDP includes one or more inequalities with large state space dimension, and a relatively small number of inputs. Preliminary numerical results illustrate that a special-purpose interior-point implementation based on these techniques can achieve a dramatic gain in efficiency compared with the best general-purpose solvers.

Several open questions remain.

- There is considerable freedom in choosing the mapping $\mathcal{L}$, used to eliminate a subset of the dual variables. The representation used in §5.1, for example, is parametrized by a state feedback matrix $K$. It is not clear how this choice affects the numerical stability of the method.
- The main idea in our approach is to use direct linear algebra techniques to solve the Newton equations in an interior-point method fast. This allows us to speed up the computation, without compromising the reliability and speed of convergence of a primal-dual interior-point method. It seems likely that other common classes of SDPs in control can benefit from similar techniques.

**Acknowledgments.**

We thank Didier Henrion, Dimitri Paucelle, Denis Arzelier, Anders Rantzer, Alexandre Megretski, Chung-Yao Kao, and Ulf Jönsson for interesting discussions on applications of KYP-SDPs and algorithms for solving them.
This material is based upon work supported by the National Science Foundation under Grant No. ECS-0200320 and the Swedish Research Council under Grant No. 271-2000-770.

## A Primal-Dual Interior-Point Method for Semidefinite Programming

In this appendix we review the definition and key properties of the semidefinite programming problem (SDP). We also describe a primal-dual method for solving SDPs.

### A.1 Optimality Conditions

We first state a few basic properties of the pair of primal and dual SDPs (21) and (22). We will express the (primal) SDP (21) as

$$\begin{array}{ll} \text{minimize} & \langle c, y\rangle \\ \text{subject to} & \mathcal{A}(y) + S + D = 0 \\ & \mathcal{B}(y) + d = 0 \\ & S \succeq 0, \end{array} \tag{61}$$

where $S \in \mathbf{S}^{l_1} \times \cdots \times \mathbf{S}^{l_L}$ is an additional variable.

The *duality gap* associated with primal feasible points $y, S$ and a dual feasible $Z$ is defined as

$$\mathbf{Tr}(SZ).$$

It is easily verified that

$$\mathbf{Tr}(SZ) = \langle c, y\rangle - \mathbf{Tr}(DZ) - d^T z$$

if $y, S, Z, z$ are primal and dual feasible. In other words the duality gap is equal to the difference between the objective values.

If strong duality holds, then $y, S, Z, z$ are optimal if and only if they are feasible, *i.e.*,

$$S \succeq 0, \qquad \mathcal{A}(y) + S + D = 0, \qquad \mathcal{B}(y) + d = 0, \tag{62}$$

and

$$Z \succeq 0, \qquad \mathcal{A}^{\mathrm{adj}}(Z) + \mathcal{B}^{\mathrm{adj}}(z) + c = 0, \tag{63}$$

and the duality gap is zero:

$$SZ = 0. \tag{64}$$

The last condition is referred to as *complementary slackness*.

### A.2 Algorithm

We briefly describe an infeasible primal-dual interior-point method for solving the pair of SDPs (61) and (22). Except for a few details, the method is the algorithm of [46, 47], which has been implemented in the state-of-the-art SDP solver SDPT3.

We assume that the mapping $(\mathcal{A}, \mathcal{B})$ has full rank, *i.e.*, $\mathcal{A}(y) = 0$ and $\mathcal{B}(y) = 0$ imply $y = 0$. We define $m = l_1 + l_2 + \cdots + l_L$.

#### Outline

The algorithm starts at initial $y$, $z$, $S$, $Z$ satisfying $S \succ 0$, $Z \succ 0$ (for example, $y = 0$, $z = 0$, $S = I$, $Z = I$). We repeat the following five steps.

1. *Evaluate stopping criteria.* Terminate if the following four conditions are satisfied:

$$\begin{aligned}
\|\mathcal{A}(y) + S + D\| &\leq \epsilon_{\text{feas}} \max\{1, \|D\|\} \\
\|\mathcal{B}(y) + d\| &\leq \epsilon_{\text{feas}} \max\{1, \|d\|\} \\
\|\mathcal{A}^{\text{adj}}(Z) + \mathcal{B}^{\text{adj}}(z) + c\| &\leq \epsilon_{\text{feas}} \max\{1, \|c\|\} \\
\mathbf{Tr}(SZ) &\leq \max\{\epsilon_{\text{abs}}, -\epsilon_{\text{rel}}\langle c, y\rangle, \epsilon_{\text{rel}}(\mathbf{Tr}(DZ) + d^T z)\},
\end{aligned}$$

   where $\epsilon_{\text{feas}}$, $\epsilon_{\text{abs}}$, $\epsilon_{\text{rel}}$ are given positive tolerances, or if a specified maximum allowable number of iterations is reached. Otherwise go to step 2.
2. *Compute the scaling matrix $R$.* The scaling matrix is a block-diagonal matrix, and defines a congruence that jointly diagonalizes $S^{-1}$ and $Z$:

$$R^T S^{-1} R = \mathbf{diag}(\lambda)^{-1}, \qquad R^T Z R = \mathbf{diag}(\lambda) \tag{65}$$

   where $\lambda \in \mathbf{R}^m_{++}$.
3. *Compute the affine scaling directions $\Delta y^{\text{a}}$, $\Delta S^{\text{a}}$, $\Delta Z^{\text{a}}$, $\Delta z^{\text{a}}$,* by solving the set of linear equations

$$\begin{aligned}
\mathcal{H}(\Delta Z^{\text{a}} S + Z \Delta S^{\text{a}}) &= -\mathbf{diag}(\lambda)^2 && (66)\\
\Delta S^{\text{a}} + \mathcal{A}(\Delta y^{\text{a}}) &= -(\mathcal{A}(y) + S + D) && (67)\\
\mathcal{A}^{\text{adj}}(\Delta Z^{\text{a}}) + \mathcal{B}^{\text{adj}}(\Delta z^{\text{a}}) &= -(\mathcal{A}^{\text{adj}}(Z) + \mathcal{B}^{\text{adj}}(z) + d) && (68)\\
\mathcal{B}(\Delta y^{\text{a}}) &= -(\mathcal{B}(y) + d), && (69)
\end{aligned}$$

   where $\mathcal{H}$ is defined as

$$\mathcal{H}(X) = \frac{1}{2}(R^T X R^{-T} + R^{-1} X^T R).$$

4. *Compute the centering-corrector steps $\Delta y^{\text{c}}$, $\Delta Z^{\text{c}}$, $\Delta S^{\text{c}}$,* by solving the set of linear equations

$$\mathcal{H}(\Delta Z^{\rm c} S + Z\Delta S^{\rm c}) = \mu I - \mathcal{H}(\Delta Z^{\rm a}\Delta S^{\rm a}) \tag{70}$$
$$\Delta S^{\rm c} + \mathcal{A}(\Delta y^{\rm c}) = 0 \tag{71}$$
$$\mathcal{A}^{\rm adj}(\Delta Z^{\rm c}) + \mathcal{B}^{\rm adj}(\Delta z^{\rm c}) = 0 \tag{72}$$
$$\mathcal{B}(\Delta y^{\rm c}) = 0. \tag{73}$$

The coefficient $\mu$ is given by

$$\mu = \frac{\mathbf{Tr}(SZ)}{m}\left(\frac{\mathbf{Tr}((S+\alpha\Delta S^{\rm a})(Z+\beta\Delta Z^{\rm a}))}{\mathbf{Tr}(SZ)}\right)^{\delta},$$

where

$$\alpha = \min\{1, \sup\{\alpha \mid S + \alpha\Delta S^{\rm a} \succeq 0\}\}$$
$$\beta = \min\{1, \sup\{\beta \mid Z + \beta\Delta Z^{\rm a} \succeq 0\}\}$$

and $\delta$ is an algorithm parameter. Typical values of $\delta$ are $\delta = 1, 2, 3$.

5. *Update the primal and dual variables* as

$$y := y+\alpha\Delta y, \qquad S := S+\alpha\Delta S, \qquad Z := Z+\beta\Delta Z, \qquad z := z+\beta\Delta z,$$

where $\Delta y = \Delta y^{\rm a} + \Delta y^{\rm c}$, $\Delta S = \Delta S^{\rm a} + \Delta S^{\rm c}$, $\Delta Z = \Delta Z^{\rm a} + \Delta Z^{\rm c}$, $\Delta z = \Delta z^{\rm a} + \Delta z^{\rm c}$, and

$$\alpha = \min\{1, 0.99\sup\{\alpha \mid S + \alpha\Delta S \succeq 0\}\}$$
$$\beta = \min\{1, 0.99\sup\{\beta \mid Z + \beta\Delta Z \succeq 0\}\}.$$

Go to step 1.

**Discussion**

**Starting Point**

The method is called *infeasible* because it does not require feasible starting points. The initial values of $y$, $z$, $S$, $Z$ must satisfy $S \succ 0$ and $Z \succ 0$, but do not have to satisfy the linear equations $\mathcal{A}(y) + S + D = 0$, $\mathcal{B}(y) + d = 0$, $\mathcal{A}^{\rm adj}(Z) + \mathcal{B}^{\rm adj}(z) + c = 0$.

The update rule in Step 5 ensures that $S \succ 0$, $Z \succ 0$ throughout the algorithm. If started at a feasible starting point, the iterates in the algorithm will remain feasible. This is easily verified from the definition of the search directions in Steps 3 and 4.

**Termination**

If we start at feasible points, the iterates satisfy $\mathcal{A}(y) + D + S = 0$, $\mathcal{B}(y) + d = 0$, $\mathcal{A}^{\rm adj}(Z) + \mathcal{B}^{\rm adj}(z) + c = 0$ throughout the algorithm, so the first

three conditions are automatically satisfied. If we start at infeasible points, these conditions ensure that at termination the primal and dual residuals $\mathcal{A}(y)+D+S$, $\mathcal{B}(y)+d$, $\mathcal{A}^{\rm adj}(Z)+\mathcal{B}^{\rm adj}(z)+c$ are sufficiently small. A typical value for $\epsilon_{\rm feas}$ is $10^{-8}$.

At each iteration, we have $S \succ 0$, $Z \succ 0$, and hence $\mathbf{Tr}(SZ) > 0$. The fourth condition is therefore satisfied if one of the following conditions holds:

- $\mathbf{Tr}(SZ) \leq \epsilon_{\rm abs}$
- $\langle c, y\rangle \leq 0$ and $\mathbf{Tr}(SZ) \leq \epsilon_{\rm rel}|\langle c, y\rangle|$
- $\mathbf{Tr}(DZ) + d^T z > 0$ and $\mathbf{Tr}(SZ) \leq \epsilon_{\rm rel}(\mathbf{Tr}(DZ) + d^T z)$.

Assuming $y$, $S$, $z$, $Z$ are feasible, the first of these conditions implies that the duality gap is less than $\epsilon_{\rm abs}$, and therefore

$$\langle c, y\rangle - p^\star \leq \epsilon_{\rm abs}, \qquad d^\star - \mathbf{Tr}(DZ) - d^T z \leq \epsilon_{\rm abs},$$

*i.e.*, the absolute errors between the primal and dual objective values and their optimal values $p^\star$ and $d^\star$ are less than $\epsilon_{\rm abs}$.

If either the second or the third condition holds, then

$$\frac{\langle c, y\rangle - p^\star}{|p^\star|} \leq \epsilon_{\rm rel}, \qquad \frac{d^\star - \mathbf{Tr}(DZ) - d^T z}{|d^\star|} \leq \epsilon_{\rm rel},$$

*i.e.*, we have determined the optimal values with a relative accuracy of at least $\epsilon_{\rm abs}$. Typical values of $\epsilon_{\rm abs}$, $\epsilon_{\rm rel}$ are $\epsilon_{\rm abs} = \epsilon_{\rm rel} = 10^{-8}$.

### Scaling Matrix

The scaling matrix $R$ is efficiently computed as follows. We first compute the Cholesky factorization of $S$ and $Z$:

$$S = L_1 L_1^T, \qquad Z = L_2 L_2^T,$$

where $L_1$ and $L_2$ are block-diagonal with lower-triangular diagonal blocks of dimensions $m_1, \ldots, m_L$. Next, we compute the SVD of $L_2^T L_1$:

$$L_2^T L_1 = U\,\mathbf{diag}(\lambda) V^T,$$

where $U$ and $V$ are block-diagonal with block dimensions $l_1, \ldots, l_L$, $U^T U = I$, $V^T V = I$, and $\mathbf{diag}(\lambda)$ is a positive diagonal matrix of size $m \times m$. Finally, we form

$$R = L_1 V\,\mathbf{diag}(\lambda)^{-1/2}.$$

It is easily verified that $R^T S^{-1} R = \mathbf{diag}(\lambda)^{-1}$ and $R^T Z R = \mathbf{diag}(\lambda)$.

**Search Directions**

The definition of $\mathcal{H}$ and the definition of the affine scaling and centering-corrector directions may be justified as follows. The *central path* for the pair of primal and dual SDPs (61) and (22) is defined as the set of points $(y(\mu), S(\mu), Z(\mu))$ that satisfy $S(\mu) \succ 0$, $Z(\mu) \succ 0$, and

$$\begin{gathered} \mathcal{A}(y(\mu)) + S(\mu) + D = 0, \qquad \mathcal{B}(y(\mu)) + d = 0 \\ \mathcal{A}^{\mathrm{adj}}(Z(\mu)) + \mathcal{B}^{\mathrm{adj}}(z(\mu)) + c = 0 \\ Z(\mu)S(\mu) = \mu I, \end{gathered} \tag{74}$$

for some $\mu > 0$. In the limit for $\mu \to 0$, these equations reduce to the optimality conditions (62)–(64). Central points with parameter $\mu$ have duality gap $\mathbf{Tr}(S(\mu)Z(\mu)) = m\mu$. Most interior-point methods can be interpreted as damped Newton methods for solving a *symmetrized* version of the central-path equations (74), for a decreasing sequence of values of $\mu$.

A unified description of different symmetric formulations of the central path was developed by Zhang [55], who notes that positive definite matrices $S$, $Z$ satisfy $SZ = \mu I$ if and only if there exists a nonsingular matrix $P$ such that

$$\frac{1}{2}(P^T Z S P^{-T} + P^{-1} S Z P) = \mu I.$$

The algorithm outlined above uses $P = R$ defined in Step 2 (known as the *Nesterov-Todd* scaling matrix), but many other choices are possible.

Using Zhang's parametrization, the central path equations can be expressed as

$$\begin{aligned} \mathcal{H}(Z(\mu)S(\mu)) &= \mu I \\ S(\mu) + \mathcal{A}(y(\mu)) + D &= 0 \\ \mathcal{A}^{\mathrm{adj}}(Z(\mu)) + \mathcal{B}^{\mathrm{adj}} z + c &= 0 \\ \mathcal{B}(y(\mu)) + d &= 0. \end{aligned}$$

The Newton directions at some $y$, $Z \succ 0$, $S \succ 0$ are obtained by linearizing these equations and solving the linearized equations

$$\mathcal{H}(\Delta Z S + Z \Delta S) = \mu I - \mathcal{H}(ZS) \tag{75}$$

$$\Delta S + \mathcal{A}(\Delta y) = -(\mathcal{A}(y) + S + D) \tag{76}$$

$$\mathcal{A}^{\mathrm{adj}}(\Delta Z) + \mathcal{B}^{\mathrm{adj}}(\Delta z) = -(\mathcal{A}^{\mathrm{adj}}(Z) + \mathcal{B}^{\mathrm{adj}}(z) + c) \tag{77}$$

$$\mathcal{B}(\Delta y) = -(\mathcal{B}(y) + d). \tag{78}$$

We can now interpret and justify the search directions defined in Steps 3, 4, and 5 as Newton directions. We first note that, if we choose $R$ as in Step 2,

$$\mathcal{H}(ZS) = \frac{1}{2}(R^T Z S R^{-T} + R^{-1} S Z R) = \mathbf{diag}(\lambda)^2,$$

so the Newton equations (75)–(78) reduce to

$$\mathcal{H}(\Delta Z S + Z \Delta S) = \mu I - \mathbf{diag}(\lambda)^2 \tag{79}$$

$$\Delta S + \mathcal{A}(\Delta y) = -(\mathcal{A}(y) + S + D) \tag{80}$$

$$\mathcal{A}^{\mathrm{adj}}(\Delta Z) + \mathcal{B}^{\mathrm{adj}}(\Delta z) = -(\mathcal{A}^{\mathrm{adj}}(Z) + \mathcal{B}^{\mathrm{adj}}(z) + c) \tag{81}$$

$$\mathcal{B}(\Delta y) = -(\mathcal{B}(y) + d). \tag{82}$$

Comparing this system with the sets of equations (66)–(69) and (70)–(73), we see that, except for the term $\mathcal{H}(\Delta Z^{\mathrm{a}} \Delta S^{\mathrm{a}})$, these equations are identical to the Newton equations. More precisely, if we delete the term $\mathcal{H}(\Delta Z^{\mathrm{a}} \Delta S^{\mathrm{a}})$, the solution (79)–(82) is given by $\Delta y = \Delta y^{\mathrm{a}} + \Delta y^{\mathrm{c}}$, $\Delta S = \Delta S^{\mathrm{a}} + \Delta S^{\mathrm{c}}$, $\Delta Z = \Delta Z^{\mathrm{a}} + \Delta Z^{\mathrm{c}}$, $\Delta z = \Delta z^{\mathrm{a}} + \Delta z^{\mathrm{c}}$.

A distinguishing feature of the predictor-corrector method is that the Newton equations are solved in two steps, by solving the two sets of linear equations (66)–(69) and (70)–(73) separately, instead of solving a single set of equations (79)–(82). This strategy has proven to be very successful in primal-dual methods for linear programming [37], and offers two advantages. The first, and most important, advantage is that it allows us to select the value of $\mu$ adaptively. In the algorithm described above, this idea is implemented as follows. In Step 3 we compute the affine scaling direction, *i.e.*, the limit of the Newton direction for $\mu \to 0$. In Step 4, we first assess the 'quality' of the affine direction as a search direction, by computing the ratio

$$\eta = \frac{\mathbf{Tr}((S + \alpha \Delta S^{\mathrm{a}})(Z + \beta \Delta Z^{\mathrm{a}}))}{\mathbf{Tr}(SZ)},$$

where we take $\alpha = 1$, $\beta = 1$ if possible, and otherwise take the maximum $\alpha$ and $\beta$ that satisfy $S + \alpha \Delta S^{\mathrm{a}} \succeq 0$, resp. $Z + \beta \Delta Z^{\mathrm{a}} \succeq 0$. The ratio $\eta$ gives the reduction in $\mathbf{Tr}(SZ)$ that we can achieve by using the affine scaling direction. If the ratio is small, we assume the affine scaling direction is a good search direction and we choose a small value of $\mu$; if the ratio $\eta$ is large, we choose a larger value of $\mu$. Choosing $\mu = \eta^{\delta} \mathbf{Tr}(SZ)/m$ means that we select the Newton step for central points $y(\mu)$, $S(\mu)$, $Z(\mu)$, with

$$\mathbf{Tr}(S(\mu)Z(\mu) = \eta^{\delta} \mathbf{Tr}(SZ).$$

The second advantage of solving two linear systems is that we can add a higher-order correction term when linearizing the equation $\mathcal{H}(Z(\mu)S(\mu)) = \mu I$. In Newton's method we linearize this equation by expanding

$$\mathcal{H}((Z + \Delta Z)(S + \Delta S)) = \mathcal{H}(ZS) + \mathcal{H}(\Delta Z S + Z \Delta S) + \mathcal{H}(\Delta Z \Delta S)$$

and omitting the second-order term, which yields a linear equation

$$\mathcal{H}(ZS) + \mathcal{H}(Z \Delta S + \Delta Z S) = \mu I.$$

The combined directions, $\Delta Z = \Delta Z^{\mathrm{a}} + \Delta Z^{\mathrm{c}}$, $\Delta S = \Delta S^{\mathrm{a}} + \Delta S^{\mathrm{c}}$, used in the predictor-corrector method, on the other hand, satisfy

$$\mathcal{H}(ZS) + \mathcal{H}((\Delta Z^{\mathrm{a}} + \Delta Z^{\mathrm{c}})S + Z(\Delta S^{\mathrm{a}} + \Delta S^{\mathrm{c}})S) + \mathcal{H}(\Delta Z^{\mathrm{a}} \Delta S^{\mathrm{a}}) = \mu I,$$

which includes part of the second-order term, and can therefore be expected to be more accurate.

We conclude by pointing out that the two sets of linear equations (66)–(69) and (70)–(73) only differ in the righthand side, so the cost of solving both systems is about the same as the cost of solving one system.

**Step Size**

After computing the search directions, we update the variables in step 5. We use different step sizes $\alpha$ and $\beta$ for the primal and dual variables. If possible, we make a full step ($\alpha = 1$, $\beta = 1$). If this is unacceptable because it results in values of $S$ and $Z$ that are not positive definite, we decrease $\alpha$ and/or $\beta$, and make a step equal to a fraction 0.99 of the maximum steps that satisfy $S + \alpha \Delta S \succeq 0$ and $Z + \beta \Delta Z \succeq 0$.

### A.3 Solving the Newton Equations

When applied to an SDP that is primal and dual feasible, the predictor-corrector method usually converges in 10–50 iterations. As a rule of thumb, the overall cost of solving the SDP and its dual is therefore equal to the cost of solving 10–50 linear equations of the form

$$\mathcal{H}(\Delta Z S + Z \Delta S) = D_1 \tag{83}$$
$$\Delta S + \mathcal{A}(\Delta y) = D_2 \tag{84}$$
$$\mathcal{A}^{\mathrm{adj}}(\Delta Z) + \mathcal{B}^{\mathrm{adj}}(\Delta z) = D_3 \tag{85}$$
$$\mathcal{B}(\Delta y) = D_4. \tag{86}$$

It can be shown that these equations have a unique solution if $(\mathcal{A}, \mathcal{B})$ has full rank [46, p.777].

The Newton equations can be simplified by eliminating $\Delta S$. Using the definition of $R$ in (65), we first note that equation (83) can be written as

$$(R^T \Delta Z R + R^{-1} \Delta S R^{-T})\, \mathbf{diag}(\lambda) + \mathbf{diag}(\lambda)(R^{-1} \Delta S R^{-T} + R^T \Delta Z R) = 2D_1.$$

The general solution of the homogeneous equation ($D_1 = 0$) is $\Delta S = -RR^T \Delta Z R R^T$. A particular solution is $\Delta Z = 0$,

$$\Delta S = 2R(D_1 \circ G)R^T$$

where $G_{ij} = 1/(\lambda_i + \lambda_j)$. All solutions of (83) can therefore be written as

$$\Delta S = -RR^T \Delta Z R R^T + 2R(D_1 \circ G)R^T.$$

Substituting in (84) gives an equivalent set of linear equations

$$-W\Delta ZW + \mathcal{A}(\Delta y) = D \tag{87}$$
$$\mathcal{A}^{\mathrm{adj}}(\Delta Z) + \mathcal{B}^{\mathrm{adj}}(\Delta z) = D_3 \tag{88}$$
$$\mathcal{B}(\Delta y) = D_4 \tag{89}$$

where $W = RR^T$, $D = D_2 - 2R(D_1 \circ G)R^T$.

A general-purpose SDP solver like SDPT3 solves (87)–(89) by eliminating $\Delta Z$ from the first equation, which yields

$$\mathcal{A}^{\mathrm{adj}}(W^{-1}\mathcal{A}(\Delta y)W^{-1}) + \mathcal{B}^{\mathrm{adj}}(\Delta z) = D_3 + \mathcal{A}^{\mathrm{adj}}(W^{-1}DW^{-1}) \tag{90}$$
$$\mathcal{B}(\Delta y) = D_4, \tag{91}$$

and solving for $\Delta y$ and $\Delta z$.

## B Derivation of (50) and (52)

### B.1 Expression for $H$

Let $H \in \mathbf{S}^{n+1}$ be defined as

$$H_{ij} = \mathbf{Tr}(F_i W F_j W), \qquad i, j = 1 \ldots, n+1,$$

where

$$F_i = \begin{bmatrix} X_i & e_i \\ e_i^T & 0 \end{bmatrix}, \quad i = 1, \ldots, n, \qquad F_{n+1} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},$$

and $X_i \in \mathbf{S}^n$. To simplify the expressions for $H$ we first note that if we partition $W$ as

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{bmatrix},$$

with $W_{11} \in \mathbf{S}^n$, $W_{12} \in \mathbf{R}^n$, and $W_{22} \in \mathbf{R}$, then

$$WF_i = \begin{bmatrix} W_{11}X_i + W_{12}e_i^T & W_{11}e_i \\ W_{12}^T X_i + W_{22}e_i^T & W_{12}^T e_i \end{bmatrix}, \quad i = 1, \ldots, n, \qquad WF_{n+1} = \begin{bmatrix} 0 & 2W_{12} \\ 0 & 2W_{22} \end{bmatrix}.$$

The leading $n \times n$ block of $H$ is given by

$$\begin{aligned} H_{ij} &= \mathbf{Tr}((W_{11}X_i + W_{12}e_i^T)(W_{11}X_j + W_{12}e_j^T)) + e_i^T W_{11}(X_j W_{12} + e_j W_{22}) \\ &\quad + (W_{12}^T X_i + W_{22}e_i^T)W_{11}e_j + e_i^T W_{12}W_{12}^T e_j \\ &= \mathbf{Tr}(W_{11}X_i W_{11}X_j) + 2e_i^T W_{11}X_j W_{12} + 2W_{12}^T X_i W_{11}e_j + 2W_{22}e_i^T W_{11}e_j \\ &\quad + 2e_i^T W_{12}W_{12}^T e_j \end{aligned}$$

for $i, j = 1, \ldots, n$. The last column is given by

$$\begin{aligned} H_{i,n+1} &= 2(W_{12}^T X_i + W_{22}e_i^T)W_{12} + 2e_i^T W_{12}W_{22} \\ &= 2W_{12}^T X_i W_{12} + 4(e_i^T W_{12})W_{22} \end{aligned}$$

for $i = 1, \ldots, n$, and

$$H_{n+1,n+1} = 4W_{22}^2.$$

In summary,

$$H = \begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix} + 2 \begin{bmatrix} W_{11} \\ W_{12}^T \end{bmatrix} \begin{bmatrix} H_2 & 0 \end{bmatrix} + 2 \begin{bmatrix} H_2^T \\ 0 \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \end{bmatrix} + 2W_{22} \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{bmatrix} + 2 \begin{bmatrix} W_{12} \\ W_{22} \end{bmatrix} \begin{bmatrix} W_{12}^T & W_{22} \end{bmatrix}$$

where

$$(H_1)_{ij} = \mathbf{Tr}(X_i W_{11} X_j W_{11}), \quad i, j = 1, \ldots, n$$
$$H_2 = \begin{bmatrix} X_1 W_{12} & X_2 W_{12} & \cdots & X_n W_{12} \end{bmatrix}.$$

This proves (50).

### B.2 Expression for $G$

Let $G \in \mathbf{R}^{(n+1)\times p}$ be defined by

$$G_{ij} = \mathbf{Tr}(F_i M_j), \qquad i = 1, \ldots, n, \quad j = 1, \ldots, p.$$

We will partition $M_j$ as

$$M_j = \begin{bmatrix} M_{j,11} & M_{j,12} \\ M_{j,12}^T & M_{j,22} \end{bmatrix},$$

with $M_{j,11} \in \mathbf{S}^n$. We have

$$G_{ij} = \mathbf{Tr}(X_i M_{j,11}) + 2e_i^T M_{j,12}, \quad i = 1, \ldots, n, \qquad G_{ij} = 2M_{j,22}.$$

From this it is easy to see that

$$G = 2 \begin{bmatrix} Y_1 B & Y_2 B & \cdots & Y_p B \\ 0 & 0 & \cdots & 0 \end{bmatrix} + 2 \begin{bmatrix} M_{1,12} & M_{2,12} & \cdots & M_{p,12} \\ M_{1,22} & M_{2,22} & \cdots & M_{p,22} \end{bmatrix},$$

with $Y_j$ is the solution of $AY_j + Y_j A^T + M_{j,11} = 0$.

## C Derivation of (53) and (54)

Let $X(v)$ be the solution of the Lyapunov equation

$$AX(v) + X(v)A^T + vB^T + Bv^T = 0,$$

*i.e.*, $X(v) = \sum_{i=1}^n v_i X_i$. The matrices $H_1$ and $H_2$ satisfy

$$v^T H_1 v = \mathbf{Tr}(X(v) W_{11} X(v) W_{11}, \qquad H_2 v = X(v) W_{12}$$

for all $v$.

### C.1 Expression for $H_1$

First suppose $A$ is diagonal, $A = \mathbf{diag}(\lambda)$, with $\lambda \in \mathbf{C}^n$. Define $\Sigma \in \mathbf{H}^{n\times n}$ as

$$\Sigma_{ij} = \frac{1}{\lambda_i + \bar{\lambda}_j}, \quad i,j = 1,\ldots,n.$$

The solution of $\mathbf{diag}(\lambda)Y + Y\,\mathbf{diag}(\lambda)^* + Gw^* + wG^* = 0$, where $G \in \mathbf{C}^n$, is given by

$$Y(w) = -(Gw^* + wG^*) \circ \Sigma. \tag{92}$$

Therefore, for general $S \in \mathbf{H}^{(n+1)\times(n+1)}$,

$$\begin{aligned}\mathbf{Tr}(Y(w)SY(w)S) &= \mathbf{Tr}(((Gw^* + wG^*) \circ \Sigma)S((Gw^* + wG^*) \circ \Sigma)S)\\ &= \mathbf{Tr}(D_G\Sigma D_w^* S D_G \Sigma D_w^* S) + \mathbf{Tr}(D_G \Sigma D_w^* S D_w \Sigma D_G^* S)\\ &\quad + \mathbf{Tr}(D_w \Sigma D_G^* S D_G \Sigma D_w^* S) + \mathbf{Tr}(D_w \Sigma D_G^* S D_w \Sigma D_G^* S)\end{aligned}$$

where $D_x = \mathbf{diag}(x)$. Now we use the property that for $A, B \in \mathbf{C}^{n\times n}$,

$$\mathbf{Tr}(D_x A D_y B) = \sum_{i=1}^n \sum_{j=1}^n x_i A_{ij} y_j B_{ji} = x^T (A \circ B^T) y.$$

This gives

$$\begin{aligned}&\mathbf{Tr}(Y(w)SY(w)S)\\ &= w^*((SD_G\Sigma) \circ (SD_G\Sigma)^T)\bar{w} + w^*(S \circ (\Sigma D_G^* S D_G \Sigma)^T)w\\ &\quad + w^T((\Sigma D_G^* S D_G \Sigma) \circ S^T)\bar{w} + w^T((\Sigma D_G^* S) \circ (\Sigma D_G^* S)^T)w\\ &= 2\,\mathrm{Re}(w^T((\Sigma D_G^* S) \circ (\Sigma D_G^* S)^T)w) + 2\,\mathrm{Re}(w^*(S \circ (\Sigma D_G^* S D_G \Sigma)^T)w). \quad (93)\end{aligned}$$

Now suppose $A$ is not diagonal, but diagonalizable, with $AV = V\,\mathbf{diag}(\lambda)$. The solution of $AX + XA^T + Bv^T + vB^T = 0$ is given by

$$X(v) = VY(V^{-1}v)V^*$$

where $Y(w)$ is the solution of

$$\mathbf{diag}(\lambda)Y + Y\,\mathbf{diag}(\lambda)^* + V^{-1}Bw^* + wB^T V^{-*} = 0.$$

Therefore, for $W_{11} \in \mathbf{S}^{n+1}$,

$$\mathbf{Tr}(X(v)W_{11}X(v)W_{11}) = \mathbf{Tr}(Y(V^{-1}v)V^*W_{11}VY(V^{-1}v)V^*W_{11}V),$$

so we can apply (93) with $w = V^{-1}v$, $S = V^*W_{11}V$, $G = V^{-1}B$, and

$$\begin{aligned}&\mathbf{Tr}(X(v)W_{11}X(v)W_{11})\\ &= 2\,\mathrm{Re}(v^T V^{-T}((\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V) \circ (\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V)^T)V^{-1}v)\\ &\quad + 2\,\mathrm{Re}(v^T V^{-*}(V^*W_{11}V) \circ (\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V\,\mathbf{diag}(V^{-1}B)\Sigma)^T)V^{-1}v).\end{aligned}$$

In conclusion,

$$\begin{aligned}H_1 = {} & 2\,\mathrm{Re}(V^{-T}\left((\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V) \circ (\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V)^T\right)V^{-1})\\ & + 2\,\mathrm{Re}(V^{-*}\left((V^*W_{11}V) \circ (\Sigma\,\mathbf{diag}(V^{-1}B)^* V^* W_{11} V\,\mathbf{diag}(V^{-1}B)\Sigma)^T\right)V^{-1}).\end{aligned}$$

### C.2 Expression for $H_2$

With $Y(w)$ defined as in (92), and $s \in \mathbf{C}^n$,

$$\begin{aligned} Y(w)s &= -((Gw^* + wG^*) \circ \Sigma)s \\ &= -D_G \Sigma D_w^* s - D_w \Sigma D_G^* s \\ &= -D_G \Sigma D_s \bar{w} - D_w \Sigma D_G^* s \\ &= -D_G \Sigma D_s \bar{w} - \mathbf{diag}(\Sigma D_G^* s) w. \end{aligned}$$

To determine $H_2$ we apply this expression with $s = V^* W_{12}$, $G = V^{-1}B$, and $w = V^{-1}v$:

$$\begin{aligned} X(v)W_{12} &= VY(V^{-1}v)V^*W_{12} \\ &= -V\,\mathbf{diag}(V^{-1}B)\Sigma\,\mathbf{diag}(V^*W_{12})\bar{V}^{-1}v \\ &\quad - V\,\mathbf{diag}(\Sigma\,\mathbf{diag}(V^{-1}B)^*V^*W_{12})V^{-1}v. \end{aligned}$$

Therefore,

$$H_2 = -V\,\mathbf{diag}(V^{-1}B)\Sigma\,\mathbf{diag}(V^*W_{12})\bar{V}^{-1} - V\,\mathbf{diag}(\Sigma\,\mathbf{diag}(V^{-1}B)^*V^*W_{12})V^{-1}.$$

## D Non-controllable $(A, B)$

In this appendix we discuss how the assumption that $(A_k, B_k)$ is controllable can be relaxed to $(A_k, B_k)$ stabilizable, provided the range of $Q_k$ is in the controllable subspace of $(A_k, B_k)$. For simplicity we explain the idea for problems with one constraint, and omit the subscripts $k$, as in (29). We define $\mathcal{M}(x) = \sum_{i=1}^p x_i M_i - N_i$, and assume the problem is strictly (primal) feasible.

Let $T$ be a unitary state transformation such that

$$\tilde{A} = \begin{bmatrix} \tilde{A}_1 & \tilde{A}_{12} \\ 0 & \tilde{A}_2 \end{bmatrix} = T^T A T, \qquad \tilde{B} = \begin{bmatrix} \tilde{B}_1 \\ 0 \end{bmatrix} = BT$$

where $(\tilde{A}_1, \tilde{B}_1)$ is controllable and $\tilde{A}_2$ is Hurwitz. Note that

$$\begin{bmatrix} T^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A^TP + PA & PB \\ B^TP & 0 \end{bmatrix} \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} & \tilde{P}\tilde{B} \\ \tilde{B}^T\tilde{P} & 0 \end{bmatrix}$$

where $\tilde{P} = T^T P T$. Let

$$\tilde{P} = \begin{bmatrix} \tilde{P}_1 & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_2 \end{bmatrix}, \qquad \tilde{Q} = \begin{bmatrix} \tilde{Q}_1 & \tilde{Q}_{12} \\ \tilde{Q}_{12}^T & \tilde{Q}_2 \end{bmatrix} = TQT^T$$

and let

$$\mathcal{M} = \begin{bmatrix} \tilde{\mathcal{M}}_1 & \tilde{\mathcal{M}}_{12} & \tilde{\mathcal{M}}_{13} \\ \tilde{\mathcal{M}}_{12}^T & \tilde{\mathcal{M}}_2 & \tilde{\mathcal{M}}_{23} \\ \tilde{\mathcal{M}}_{13}^T & \tilde{\mathcal{M}}_{23}^T & \tilde{\mathcal{M}}_3 \end{bmatrix} = \begin{bmatrix} T^T & 0 \\ 0 & I \end{bmatrix} \mathcal{M}(x) \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}.$$

Then it holds that (29) with strict inequality is equivalent to

$$\begin{array}{ll} \text{minimize} & q^T x + \mathbf{Tr}(\tilde{Q}_1 \tilde{P}) + 2\,\mathbf{Tr}(\tilde{Q}_{12} P_{12}) + \mathbf{Tr}(\tilde{Q}_2 \tilde{P}_2) \\ \text{subject to} & \begin{bmatrix} \tilde{P}_1 \tilde{A}_1 + \tilde{A}_1^T \tilde{P}_1 & \tilde{P}_1 \tilde{A}_{12} + \tilde{P}_{12} \tilde{A}_2 + \tilde{A}_1^T \tilde{P}_{12} & \tilde{P}_1 \tilde{B}_1 \\ * & \tilde{P}_{12}^T \tilde{A}_{12} + \tilde{P}_2 \tilde{A}_2 + \tilde{A}_{12}^T \tilde{P}_{12} + \tilde{A}_2^T \tilde{P}_2 & \tilde{P}_{12}^T \tilde{B}_1 \\ * & * & 0 \end{bmatrix} \\ & + \begin{bmatrix} \tilde{\mathcal{M}}_1 & \tilde{\mathcal{M}}_{12} & \tilde{\mathcal{M}}_{13} \\ \tilde{\mathcal{M}}_{12}^T & \tilde{\mathcal{M}}_2 & \tilde{\mathcal{M}}_{23} \\ \tilde{\mathcal{M}}_{13}^T & \tilde{\mathcal{M}}_{23}^T & \tilde{\mathcal{M}}_3 \end{bmatrix} \succ 0. \end{array} \tag{94}$$

By the Schur complement formula the above constraint is equivalent to

$$\begin{bmatrix} \tilde{P}_1 \tilde{A}_1 + \tilde{A}_1^T \tilde{P}_1 & \tilde{P}_1 \tilde{B}_1 \\ \tilde{B}_1^T \tilde{P}_1 & 0 \end{bmatrix} + \begin{bmatrix} \tilde{\mathcal{M}}_1 & \tilde{\mathcal{M}}_{13} \\ \tilde{\mathcal{M}}_{13}^T & \tilde{\mathcal{M}}_3 \end{bmatrix} \succ 0$$

and

$$\begin{aligned} &\tilde{P}_{12}^T \tilde{A}_{12} + \tilde{P}_2 \tilde{A}_2 + \tilde{A}_{12}^T \tilde{P}_{12} + \tilde{A}_2^T \tilde{P}_2 + \tilde{\mathcal{M}}_2 \\ &\quad - \begin{bmatrix} \tilde{P}_1 \tilde{A}_{12} + \tilde{P}_{12} \tilde{A}_2 + \tilde{A}_1^T \tilde{P}_{12} + \tilde{\mathcal{M}}_{12} \\ B_1^T \tilde{P}_{12} + \tilde{\mathcal{M}}_{23}^T \end{bmatrix}^T \begin{bmatrix} \tilde{P}_1 \tilde{A}_1 + \tilde{A}_1^T \tilde{P}_1 + \tilde{\mathcal{M}}_1 & \tilde{P}_1 \tilde{B}_1 + \tilde{\mathcal{M}}_{13} \\ \tilde{B}_1^T \tilde{P}_1 + \tilde{\mathcal{M}}_{13}^T & \tilde{\mathcal{M}}_3 \end{bmatrix}^{-1} \\ &\quad \times \begin{bmatrix} \tilde{P}_1 \tilde{A}_{12} + \tilde{P}_{12} \tilde{A}_2 + \tilde{A}_1^T \tilde{P}_{12} + \tilde{\mathcal{M}}_{12} \\ \tilde{B}_1^T \tilde{P}_{12} + \tilde{\mathcal{M}}_{23}^T \end{bmatrix} \succ 0. \end{aligned}$$

Now by our assumption that the range of $Q$ is in the controllable subspace of $(A, B)$, we have $\tilde{Q}_2 = 0$ and $\tilde{Q}_{12} = 0$. Then $\tilde{P}_{12}$ and $\tilde{P}_2$ only appear in the latter matrix inequality. This shows that it is possible to partition the optimization problem into one problem of the original form for which $(\tilde{A}_1, \tilde{B}_1)$ is controllable involving the variables $x$ and $\tilde{P}_1$, and a feasibility problem involving $\tilde{P}_{12}$ and $\tilde{P}_2$. Notice that feasible $\tilde{P}_{12}$ and $\tilde{P}_2$ can be found by solving a Lyapunov equation for $\tilde{P}_2$. Hence all results presented in this article extend to the case when $(A, B)$ is stabilizable. Notice however that there does not exist strictly dual feasible $Z$ if $(A, B)$ is not controllable.

## E Linear Independence

In this appendix, we relax the assumption that the mapping (6) has full rank.

### E.1 Change of Variables

Consider the constraint in (29) which can be written as

$$-\mathcal{A}(P, x) = \begin{bmatrix} PA + A^T P & PB \\ B^T P & 0 \end{bmatrix} + \sum_{i=1}^{p} x_i \begin{bmatrix} M_{1,i} & M_{12,i} \\ M_{12,i}^T & M_{2,i} \end{bmatrix} \succeq \begin{bmatrix} N_1 & N_{12} \\ N_{12}^T & N_2 \end{bmatrix}.$$

Let $P_i$ solve

$$A^T P_i + P_i A = M_{1,i}, \quad i = 1, \ldots, p$$

and let $\tilde{M}_{12,i} = M_{12,i} - P_i B$, $i = 1, \ldots, p$. Then with

$$\bar{P} = P - \sum_{i=1}^{p} x_i P_i$$

it holds that the above LMI is satisfied for $P = P^T$ and some $x$ if and only if $\bar{P}$ and $x$ satisfy

$$-\tilde{\mathcal{A}}(\bar{P}, x) = \begin{bmatrix} \bar{P}A + A^T\bar{P} & \bar{P}B \\ B^T\bar{P} & 0 \end{bmatrix} + \sum_{i=1}^{p} x_i \tilde{M}_i \succeq N$$

where

$$\tilde{M}_i = \begin{bmatrix} 0 & \tilde{M}_{12,i} \\ \tilde{M}_{12,i}^T & M_{2,i} \end{bmatrix}.$$

Hence it is no loss in generality to assume that the LMI constraint is of a form where the $M_{1,i}$-entries are zero. The objective function is transformed to $\tilde{q}^T x + \mathbf{Tr}\, Q\bar{P}$, where $\tilde{q}_i = q_i + \mathbf{Tr}\, QP_i$. We remark that this structure of the constraint is inherent in certain applications such as IQCs as they are defined in the Matlab IQC-toolbox. Moreover, notice that we could have defined the change of variables slightly differently using an affine change of variables such that $N$ would also have had a zero 1,1-block. However, the notation would have been more messy, and it would also complicate the presentation in what follows.

### E.2 Linear Independence

The full rank property of $\mathcal{A}(P, x)$ is needed for uniqueness of the solution of the Newton equations. In this subsection we show that a sufficient and more easily verified condition is that $A$ is Hurwitz and $\tilde{M}_i$, $i = 1, \ldots, p$, are linearly independent. We make use of the specific structure developed above to show that $\mathcal{A}(P, x) = 0$ implies $(P, x) = 0$. By the change of variables in the previous subsection, $\mathcal{A}(P, x) = 0$ is equivalent to $\tilde{\mathcal{A}}(\bar{P}, x) = 0$. In the 1,1-position this equation reads

$$\bar{P}A + A^T\bar{P} = 0$$

and since $A$ is Hurwitz it follows that $\bar{P} = 0$. This implies that

$$\sum_{i=1}^{p} x_i \tilde{M}_i = 0$$

and since $\tilde{M}_i$, $i = 1, \ldots, p$, are linearly independent it follows that $x = 0$. By the definition of the change of variables it is now true that $(P, x) = 0$. We remark that the above proof easily extends to the general problem formulation in the introduction.

### E.3 Linear Dependence

In case $\tilde{M}_i$, $i = 1, \ldots, p$, are linearly dependent, then either the objective function is not bounded from below, or the problem can be reduced to an equivalent problem with fewer variables for which $\tilde{M}_i$, $i = 1, \ldots, p$, are linearly independent. To this end define

$$\tilde{M} = \left[ \mathbf{svec}(\tilde{M}_1)\ \mathbf{svec}(\tilde{M}_2) \cdots \mathbf{svec}(\tilde{M}_p) \right].$$

Apply a singular value decomposition to $\tilde{M}$:

$$\tilde{M} = U \left[ \Sigma_1\ 0 \right] \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

where $\Sigma_1$ has full column rank. Define a change of variables for $x$ via

$$\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} x$$

Now clearly $\tilde{M}x = \bar{M}\bar{x}_1$, where $\bar{M} = U\Sigma_1$. Therefore we can rewrite the constraint in the variables $\bar{x}_1$ and with a set of linearly independent matrices, $\bar{M}_i$, given by the inverse symmetric vectorization of the columns of $\bar{M}$. The part of the primal objective function involving $x$ can be written as as

$$c^T x = c^T \left[ V_1\ V_2 \right] \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} x = \bar{c}_1^T \bar{x}_1 + \bar{c}_2^T \bar{x}_2$$

where $\bar{c}_1 = V_1^T c$, $\bar{c}_2 = V_2^T c$. Since $\bar{x}_2$ is not present in the constraint the objective function is bounded below only if

$$\bar{c}_2 = V_2^T c = 0$$

Hence, this is a necessary condition for the optimization problem to have a solution. Therefore either the problem is not bounded below or there is an equivalent problem involving fewer variables for which $\bar{M}_i$, $i = 1, \ldots, p$ are linearly independent.

The cost of the above operation is $O(n^3)$, where we assume that $p$ is $O(n)$.

## References

1. P. Apkarian and R. J. Adams (1998). Advanced gain-scheduled techniques for uncertain systems. IEEE Trans. Control Sys. Tech., 6(1):21–32.
2. P. Apkarian and P. Gahinet (1995). A convex characterization of gain-scheduled $\mathbf{H}_\infty$ controllers. IEEE Trans. Aut. Control, 40(5):853–864.
3. F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta (1997). SDPPACK User's Guide, Version 0.9 Beta. NYU.

4. B. Alkire and L. Vandenberghe (2000). Handling nonnegative constraints in spectral estimation. Proceedings of the 34th Asilomar Conference on Signals, Systems, and Computers, 202–206.
5. B. Alkire and L. Vandenberghe (2001). Interior-point methods for magnitude filter design. Proceedings of the 2001 IEEE International Conference on Acoustics. Speech, and Signal Processing.
6. B. Alkire and L. Vandenberghe (2002). Convex optimization problems involving finite autocorrelation sequences. Mathematical Programming Series A, 93:331–359.
7. S. Boyd and C. Barratt (1991). Linear Controller Design: Limits of Performance. Prentice Hall.
8. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory. Volume 15 of Studies in Applied Mathematics. SIAM, Philadelphia, PA.
9. B. Borchers (2002). CSDP 4.2 User's Guide. Available from `www.nmt.edu/~borchers/csdp.html`.
10. V. Balakrishnan and L. Vandenberghe (1998). Linear Matrix Inequalities for signal processing: An overview. Proceedings of the 32nd Annual Conference on Information Sciences and Systems, Princeton, New Jersey.
11. V. Balakrishnan and L. Vandenberghe (2003). Semidefinite programming duality and linear time-invariant systems. IEEE Trans. Aut. Control, 48:30–41.
12. V. Balakrishnan and F. Wang (1999). Efficient computation of a guaranteed lower bound on the robust stability margin for a class of uncertain systems. IEEE Trans. Aut. Control, AC-44(11):2185–2190.
13. S. J. Benson and Y. Ye (2002). DSDP4 — A Software Package Implementing the Dual-Scaling Algorithm for Semidefinite Programming. Available from `www-unix.mcs.anl.gov/~benson`.
14. T. N. Davidson, Z.-Q. Luo, and J. F. Sturm (2002). Linear matrix inequality formulation of spectral mask constraints with applications to FIR filter design. IEEE Transactions on Signal Processing, 50(11):2702–2715.
15. J. Doyle (1982). Analysis of feedback systems with structured uncertainties. IEE Proc., 129-D(6):242–250.
16. B. Dumitrescu, I. Tabus, and P. Stoica (2001). On the parametrization of positive real sequences and MA parameter estimation. IEEE Transactions on Signal Processing, 49(11):2630–9.
17. M. Fu and N. E. Barabanov (1997). Improved Upper Bounds for the Mixed Structured Singular Value. IEEE Trans. Aut. Control, 42(10):1447–1452.
18. K. Fujisawa, M. Kojima, and K. Nakata (1998). SDPA User's Manual. Available from `www.is.titech.ac.jp/~yamashi9/sdpa`.
19. M. K. H. Fan, A. L. Tits, and J. C. Doyle (1991). Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. IEEE Trans. Aut. Control, AC-36(1):25–38.
20. J. Gillberg and A. Hansson (2003). Polynomial complexity for a Nesterov-Todd potential-reduction method with inexact search directions. Proc. IEEE Conference on Decision and Control.
21. Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren (2000). Convex optimization over positive polynomials and filter design. Proc. UKACC International Conference on Control, Cambridge University.

22. Y. Genin, Y. Hachez, Yu. Nesterov, and P. Van Dooren (2003). Optimization problems over positive pseudo-polynomial matrices. SIAM Journal on Matrix Analysis and Applications, 25(3):57–79.
23. M. Green and D. J. N. Limebeer (1995). Linear Robust Control. Information and System sciences. Prentice Hall, Englewood Cliffs, NJ.
24. P. Gahinet and A. Nemirovskii (1995). LMI Control Toolbox for Matlab. The MathWorks, Inc.
25. Y. Hachez (2003). Convex Optimization over Non-Negative Polynomials: Structured Algorithms and Applications. PhD thesis, Université Catholique de Louvain, Belgium.
26. H. Hindi and S. Boyd (1999). Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$-optimal control via finite-dimensional $Q$-parametrization and linear matrix inequalities. Proc. American Control Conf.
27. D. Henrion (2003). LMI formulations of polynomial matrix problems in robust control. Draft.
28. A. Hansson and L. Vandenberghe (2000). Efficient solution of linear matrix inequalities for integral quadratic constraints. Proc. IEEE Conf. on Decision and Control, 5033–5034.
29. A. Hansson and L. Vandenberghe (2001). A primal-dual potential reduction method for integral quadratic constraints. Proc. American Control Conference, 3013–3018, Arlington, Virginia.
30. T. Iwasaki and S. Hara (1998). Well-posedness of feedback systems: Insights into exact robustness analysis and approximate computations. IEEE Trans. Aut. Control, AC-43(5):619–630.
31. U. Jönsson (1996). Robustness Analysis of Uncertain and Nonlinear Systems. PhD thesis, Lund Institute of Technology, Sweden.
32. C.-Y. Kao and A. Megretski (2001). Fast algorithms for solving IQC feasibility and optimization problems. Proc. American Control Conf., 3019–3024.
33. C.-Y. Kao, A. Megretski, and U. T. Jönsson (2001). A cutting plane algorithm for robustness analysis of periodically time-varying systems. IEEE Trans. Aut. Control, 46(4):579–592.
34. C.-Y. Kao, A. Megretski, and U. Jönsson (2003). Specialized fast algorithms for IQC feasibility and optimization problems. Submitted for publication.
35. J. Kautsky, N. K. Nichols, and P. Van Dooren (1985). Robust pole assignment in linear state feedback. Int. J. Control, 41:1129–1155.
36. J. Löfberg (2002). YALMIP. Yet another LMI parser. University of Linköping, Sweden.
37. S. Mehrotra (1991). On the implementation of a primal-dual interior point method. SIAM Journal on Optimization, 2(4):575–601.
38. A. Megretski and A. Rantzer (1997). System analysis via integral quadratic constraints. IEEE Trans. Aut. Control, 42(6):819–830.
39. J. Oishi and V. Balakrishnan (1999). Linear controller design for the NEC laser bonder via LMI optimization. In Laurent El Ghaoui and Silviu-Iulian Niculescu (Editors). Advances in Linear Matrix Inequality Methods in Control. Advances in Control and Design. SIAM.
40. A. Packard (1994). Gain scheduling via linear fractional transformations. Syst. Control Letters, 22:79–92.
41. P. A. Parrilo (2000). Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. PhD thesis, California Institute of Technology, Pasadena, California.

42. A. Rantzer (1996). On the Kalman-Yakubovich-Popov lemma. Syst. Control Letters, 28(1):7–10.
43. M. G. Safonov (1982). Stability margins of diagonally perturbed multivariable feedback systems. IEE Proc., 129-D:251–256.
44. J. F. Sturm (2001). Using SEDUMI 1.02, a Matlab Toolbox for Optimization Over Symmetric Cones. Available from `fewcal.kub.nl/sturm/software/sedumi.html`.
45. J. F. Sturm (2002). Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Optimization Methods and Software, 17(6):1105–1154.
46. M. J. Todd, K. C. Toh, and R. H. Tütüncü (1998). On the Nesterov-Todd direction in semidefinite programming. SIAM J. on Optimization, 8(3):769–796.
47. K. C. Toh, R. H. Tütüncü, and M. J. Todd (2002). SDPT3 version 3.02. A Matlab software for semidefinite-quadratic-linear programming. Available from `www.math.nus.edu.sg/~mattohkc/sdpt3.html`.
48. S.-P. Wu and S. Boyd (1996). SDPSOL: A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Version Beta. Stanford University.
49. F. Wang and V. Balakrishnan (2002). Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems. IEEE Trans. Aut. Control, 47(5):720–734.
50. S.-P. Wu, S. Boyd, and L. Vandenberghe (1998). FIR filter design via spectral factorization and convex optimization. In B. Datta (Editor). Applied and Computational Control, Signals, and Circuits, 1:215–245, Birkhauser.
51. F. Wang, V. Balakrishnan, P. Zhou, J. Chen, R. Yang, and C. Frank (2003). Optimal array pattern synthesis using semidefinite programming. IEEE Trans. Signal Processing, 51(5):1172–1183.
52. R. Wallin, H. Hansson, and L. Vandenberghe (2003). Comparison of two structure-exploiting optimization algorithms for integral quadratic constraints. Proc. IFAC Symposium on Robust Control Design, Milan, Italy.
53. J. C. Willems (1971). Least squares stationary optimal control and the algebraic Riccati equation. IEEE Trans. Aut. Control, AC-16(6):621–634.
54. K. Zhou, J. Doyle, and K. Glover (1996). Robust and Optimal Control. Prentice Hall.
55. Y. Zhang (1998). On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming. SIAM J. on Optimization, 8:365–386.

# Optimization Problems over Non-negative Polynomials with Interpolation Constraints

Yvan Hachez[1] and Yurii Nesterov[2]

[1] Work carried out at Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Belgium.
Present address: Electrabel, B-1000 Brussels, Belgium.
[2] Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Belgium.

Optimization problems over several cones of non-negative polynomials are described; we focus on linear constraints on the coefficients that represent interpolation constraints. For these problems, the complexity of solving the dual formulation is shown to be almost independent of the number of constraints, provided that an appropriate preprocessing has been performed. These results are also extended to non-negative matrix polynomials and to interpolation constraints on the derivatives.

## 1 Introduction

Non-negative polynomials are natural objects to model various engineering problems. Among the most representative applications are the filter design problems [1, 2, 5]. Recently, self-concordant barriers for several cones of non-negative polynomials have been proposed in the literature [12]. They are usually based on results dating back to the beginning of the *20*th century [10]. Indeed, these cones and their properties were extensively studied by several well-known mathematicians (Fejér, Riesz, Toeplitz, Markov, ...), as testified by their correspondences and papers.

Nowadays, convex optimization techniques allow us to efficiently treat these cones, which are parametrized by semidefinite matrices [13, 16]. Although general semidefinite programming solvers could be used to solve the associated problems, the inherent structure of these polynomial problems must be exploited to derive much more efficient algorithms [1, 6, 7]. They are based on the matrix structure that shows up in the dual problem. In particular, solving the standard conic formulation on cones of non-negative polynomials using the dual matrix structure has been studied in [6].

In this chapter, we consider particular conic formulations, of which the linear constraints are interpolations constraints. Indeed, natural linear con-

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 239–271, 2005.

straints on the coefficients of a polynomial are obtained as interpolation conditions on the polynomial or its derivatives; each of them has an unambiguous interpretation. We show that the associated optimization problems can be solved very efficiently in a number of flops almost independent of the polynomial degree. Moreover, these formulations have some interesting properties that are worth pointing out.

In Section 2, we remind the reader of the characterization of non-negative scalar polynomials using the cone of positive semidefinite matrices. This step is of paramount importance before formulating the conic convex problems of interest, i.e., minimizing a linear function of the coefficients of a non-negative polynomial subject to interpolation constraints, see Section 3. Under mild assumptions, these problems can be solved very efficiently as described in Section 4. Although non-negative matrix polynomials can also be characterized using positive semidefinite matrices, closed formulas and nice interpretations are more difficult to obtain. Section 5 shows how to extend our previous results to this new setting. In Section 6, interpolation constraints on the derivatives are considered. Although we could have started this paper with the general setting (non-negative matrix polynomials with general interpolation constraints), this general approach and the associated notation would have shadowed most of the basic ideas underlying our results.

*Notation*

The optimization problems considered hereafter are assumed to be stated in terms of appropriate scalar products defined over the space of complex matrices. For any couple of matrices $X$ and $Y$, let us set their Frobenius scalar product as follows

$$\langle X, Y\rangle_F \doteq \mathrm{Re}(\mathrm{Trace}\, XY^*) \equiv \mathrm{Re}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{i,j}\overline{y}_{i,j}, \tag{1}$$

where $\{x_{i,j}\}_{i,j}$ and $\{y_{i,j}\}_{i,j}$ are the scalar entries of the matrices $X$ and $Y$, respectively. Both matrices must have the same dimension $m \times n$, but they are not necessarily square. The above definition can thus be applied to vectors. Since this scalar product induces the Frobenius norm, i.e. $\|X\|_F^2 = \langle X, X\rangle_F$, it is called the Frobenius scalar product. It also follows from the definition that

$$\langle X, Y\rangle_F = \langle \mathrm{Re}(X), \mathrm{Re}(Y)\rangle + \langle \mathrm{Im}(X), \mathrm{Im}(Y)\rangle$$

where $\langle \cdot, \cdot \rangle$ stands for the standard scalar product of matrices, i.e. $\langle X, Y\rangle = \mathrm{Trace}\, XY^*$. Positive semidefiniteness of a matrix $Y$ is denoted by $Y \succeq 0$. The sets of positive semidefinite real symmetric and complex Hermitian matrices (of order $n$) are denoted by $\mathcal{S}_n^+$ and $\mathcal{H}_n^+$, respectively. The column vector

$$\pi_n(s) = \begin{bmatrix} 1 & s & \cdots & s^n \end{bmatrix}^T,$$

with $s = x$ or $z$, is often used to represent a polynomial by its coefficients. The (block) diagonal matrix defined by the (block) vector $y$ is denoted by $\mathrm{diag}(y)$. The complex unit is written as i, i.e. $\mathrm{i}^2 = -1$. The elements of the canonical basis are written as $\{e_k\}_k$, i.e. $I_n = \begin{bmatrix} e_0 & \dots & e_{n-1} \end{bmatrix}$ is the identity matrix of size $n$. $\binom{n}{k}$ is the binomial coefficient $\frac{n!}{(n-k)!k!}$.

## 2 Non-negative Polynomials

Let us summarize a few facts about non-negative polynomials. First of all, the characterization of such polynomials depends on the curve of the complex plane on which they are defined. These curves are typically the real axis $\mathbb{R}$, the unit circle $\mathrm{e}^{\mathrm{i}\mathbb{R}}$ or the imaginary axis $\mathrm{i}\mathbb{R}$. Optimization problems on the latter curve are not considered in what follows; they can be reduced in a straightforward manner to optimization problems on the real line. The set of non-negative polynomials on any of these three curves is clearly a convex cone $\mathcal{K}$, i.e.

$$\mathcal{K} + \mathcal{K} \subseteq \mathcal{K}, \quad \alpha\mathcal{K} \subseteq \mathcal{K}, \quad \forall \alpha \geq 0$$

In this article, this special structure is used to formulate various optimization problems in conic form, based on interpolation constraints. Let us now examine the cones of interest and their duals.

### 2.1 Real Line

Denote the cone of polynomials (of degree $2n$) non-negative on the whole real line $\mathbb{R}$ by

$$\mathcal{K}_{\mathbb{R}} = \{p \in \mathbb{R}^{2n+1} : p(x) = \sum_{k=0}^{2n} p_k x^k \geq 0, \forall x \in \mathbb{R}\}$$

and define the inner product between two real vectors $p = \begin{bmatrix} p_0, \dots, p_{2n} \end{bmatrix}^T$ and $q = \begin{bmatrix} q_0, \dots, q_{2n} \end{bmatrix}^T$ by $\langle p, q \rangle_{\mathbb{R}} \doteq \sum_{k=0}^{2n} p_k q_k = \langle p, q \rangle$. As a direct consequence of Markov-Lukács Theorem , the cone $\mathcal{K}_{\mathbb{R}}$ can be characterized as follows [12].

**Theorem 1.** *A polynomial $p(x) = \sum_{k=0}^{2n} p_k x^k$ is non-negative on the real axis if and only if there exists a positive semidefinite symmetric matrix $Y = \{y_{ij}\}_{i,j=0}^{n}$ such that ($y_{ij} = 0$ for $i$ or $j$ outside their definition range)*

$$p_k = \sum_{i+j=k} y_{ij}, \quad \text{for } k = 0, \dots, 2n. \tag{2}$$

*Remark 1.* Identities (2) can be rewritten as $p(x) = \langle Y \pi_n(x), \pi_n(x) \rangle$.

Remember that the dual cone $\mathcal{K}_{\mathbb{R}}^{\star}$ is defined by

$$\mathcal{K}_{\mathbb{R}}^{\star} = \{s \in \mathbb{R}^{2n+1} : \langle p, s\rangle_{\mathbb{R}} \geq 0, \quad \forall p \in \mathcal{K}_{\mathbb{R}}\}.$$

Let $H(s)$ be the *Hankel matrix* defined by the vector $s \in \mathbb{R}^{2n+1}$, i.e.,

$$H(s) = \begin{bmatrix} s_0 & s_1 & \cdots & s_n \\ s_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & s_{2n-1} \\ s_n & \cdots & s_{2n-1} & s_{2n} \end{bmatrix}. \tag{3}$$

Then the cone dual to $\mathcal{K}_{\mathbb{R}}$ is characterized by $H(s) \succeq 0$, i.e.,

$$\mathcal{K}_{\mathbb{R}}^{\star} = \{s \in \mathbb{R}^{2n+1} : H(s) \succeq 0\}.$$

The operator dual to $H(\cdot)$ allows us to write (2) as $p = H^*(Y)$, which means that

$$p_k = \langle Y, H(e_k)\rangle, \quad k = 0, \ldots, 2n.$$

Let us now interpret the primal and dual objects. Given $p \in \operatorname{int} \mathcal{K}_{\mathbb{R}}$, there exists a positive definite matrix $Y$ such that $p = H^*(Y)$ and such that its inverse is a Hankel matrix, say $Y = H(s)^{-1}$ [12]. Remember that any positive definite Hankel matrix $H(s)$ admits a Vandermonde factorization [3]

$$Y^{-1} = H(s) = V \operatorname{diag}(\{p_i\}_{i=0}^n)^{-1} V^T \tag{4}$$

where $p_i > 0, \forall i$ and $V$ is a nonsingular Vandermonde matrix defined by the nodes $\{x_i\}_{i=0}^n$, i.e.

$$V = \begin{bmatrix} 1 & \ldots & 1 \\ x_0 & \ldots & x_n \\ \vdots & & \vdots \\ x_0^n & \ldots & x_n^n \end{bmatrix}. \tag{5}$$

Let $L^T$ be the inverse of $V$, i.e. $L^T V = I_{n+1}$. It is well known that the rows of $L^T$ are the coefficients of the *Lagrange polynomials* $\{l_i(x)\}_{i=0}^n$ associated with the distinct points $\{x_i\}_{i=0}^n$ :

$$l_i(x_j) = \langle L e_i, \pi_n(x_j)\rangle = \delta_{ij}, \quad 0 \leq i, j \leq n,$$

where $\delta_{ij}$ is the Kronecker delta. Therefore, we obtain an explicit expression of the positive definite matrix $Y$ in terms of the Lagrange polynomials

$$Y = L \operatorname{diag}(\{p_i\}_{i=0}^n) L^T. \tag{6}$$

This representation is equivalent to the decomposition of $p(x)$ as a weighted sum of $n+1$ squared Lagrange polynomials

$$p(x) = \sum_{i=0}^{n} p(x_i)(l_i(x))^2.$$

To see this, we plug expression (6) into the identity $p(x) = \langle Y\pi_n(x), \pi_n(x)\rangle$, which can subsequently be rewritten as

$$\begin{aligned} p(x) &= \langle L\,\mathrm{diag}(\{p_i\}_{i=1}^{n+1})L^T\pi_n(x), \pi_n(x)\rangle = \sum_{i=0}^{n} p_i\langle Le_ie_i^TL^T, \pi_n(x)\pi_n(x)^T\rangle \\ &= \sum_{i=0}^{n} p_i(\langle Le_i, \pi_n(x)\rangle)^2 = \sum_{i=0}^{n} p_i(l_i(x))^2 \end{aligned}$$

As $\{l_i(x)\}_{i=0}^n$ are the Lagrange polynomials associated with the points $\{x_i\}_{i=0}^n$, it is straightforward to check that $p_i = p(x_i), \forall i$.

### 2.2 Unit Circle

On the unit circle, the non-negative polynomials of interest are the trigonometric polynomials. Remember that a trigonometric polynomial of degree $n$ has the form

$$p(\theta) = \sum_{k=0}^{n}[a_k\cos(k\theta) + b_k\sin(k\theta)], \quad \theta \in [0, 2\pi]. \tag{7}$$

where $\{a_k\}_{k=0}^n$ and $\{b_k\}_{k=0}^n$ are two sets of real coefficients. Without loss of generality, we can assume that $b_0 = 0$.

If we define the complex coefficients $\{p_k\}_{k=0}^n$ as

$$p_k = a_k + \mathrm{i}b_k, \quad k = 0, \ldots, n,$$

the pseudo-polynomial

$$p(z) = \langle p, \pi_n(z)\rangle_F = \mathrm{Re}(\sum_{k=0}^{n} p_k z^{-k}), \quad |z| = 1, \tag{8}$$

evaluated on the unit circle is equivalent to trigonometric polynomial (7). Therefore, we can use either (7) or (8) to represent the *same* mathematical object.

Denote the cone of trigonometric polynomials (of degree $n$) non-negative on the unit circle by

$$\mathcal{K}_{\mathbb{C}} = \{p \in \mathbb{R}\times\mathbb{C}^n : \langle p, \pi_n(z)\rangle_F \geq 0, z = \mathrm{e}^{\mathrm{i}\theta}, \theta \in [0, 2\pi)\}.$$

and define the inner product between two vectors $p = \begin{bmatrix} p_0, \ldots, p_n \end{bmatrix}^T \in \mathbb{R}\times\mathbb{C}^n$ and $q = \begin{bmatrix} q_0, \ldots, q_n \end{bmatrix}^T \in \mathbb{R}\times\mathbb{C}^n$ by $\langle p, q\rangle_{\mathbb{C}} \doteq \langle p, q\rangle_F$. As a direct consequence of Fejér-Riesz Theorem , see e.g., [15, Part 6, Problems 40 and 41], this cone can be characterized as follows [12].

**Theorem 2.** *A trigonometric polynomial $p(z) = \langle p, \pi_n(z)\rangle_F$ is non-negative on the unit circle if and only if there exists a positive semidefinite Hermitian matrix $Y = \{y_{i,j}\}_{i,j=0}^n$ such that ($y_{i,j} = 0$ for $i$ or $j$ outside their definition range)*

$$p_k = \begin{cases} \sum_{i-j=0} y_{i,j}, & k = 0 \\ 2\sum_{i-j=k} y_{i,j}, & k = 1, \ldots, n \end{cases} \tag{9}$$

*Remark 2.* As before, identities (9) can be rewritten using the vector $\pi_n(z)$, i.e. $p(z) = \langle Y\pi_n(z), \pi_n(z)\rangle$.

By definition, the cone dual to $\mathcal{K}_\mathbb{C}$ is the set of vectors $s \in \mathbb{R}\times\mathbb{C}^n$ satisfying the inequalities

$$\langle p, s\rangle_\mathbb{C} \geq 0, \quad \forall p \in \mathcal{K}_\mathbb{C}.$$

Let $T(s)$ be the Hermitian Toeplitz matrix defined by the vector $s \in \mathbb{R} \times \mathbb{C}^n$, i.e.,

$$T(s) = \begin{bmatrix} s_0 & \bar{s}_1 & \cdots & \bar{s}_n \\ s_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{s}_1 \\ s_n & \ddots & s_1 & s_0 \end{bmatrix}. \tag{10}$$

Then the cone dual to $\mathcal{K}_\mathbb{C}$ is characterized by $T(s) \succeq 0$, i.e.

$$\mathcal{K}_\mathbb{C}^\star = \{s \in \mathbb{R} \times \mathbb{C}^n : T(s) \succeq 0\}.$$

The operator dual to $T(\cdot)$ allows us to write (9) as $p = T^*(Y)$, which means that

$$p_k = \langle Y, T_k\rangle, \quad k = 0, \ldots, n$$

where the matrices $\{T_k\}_{k=0}^n$ are defined by the identity $T(s) = \frac{1}{2}\sum_{k=0}^n (T_k s_k + T_k^T \overline{s}_k), \forall s \in \mathbb{R} \times \mathbb{C}^n$.

As before, a better understanding of the primal and dual objects is obtained by considering the decomposition of $p(z) \in \operatorname{int}\mathcal{K}_\mathbb{C}$ as a weighted sum of squared Lagrange polynomials.

## 3 The Optimization Problem

The problem of optimizing over the cone of non-negative polynomials, subject to linear constraints on the coefficients of these polynomials, has already been studied by the authors in a wider framework [6]. Remember that this class of problems is exactly the standard *conic formulation* introduced in [13]. In this section, we now focus on the particular case of scalar polynomials constrained by interpolation constraints. The consequent structures of the primal and dual problems lead to efficient algorithms for solving such problems.

### 3.1 Real Line

Several important optimization problems on the real line can be formulated as the following *primal* problem

$$\begin{array}{l} \min \langle c, p\rangle \\ \text{s.t. } \langle a_i, p\rangle = b_i,\ i = 0, \ldots, k-1, \\ \quad p \in \mathcal{K}_{\mathbb{R}}, \end{array} \tag{11}$$

where the matrix of constraints $A = \{a_i\}_{i=0}^{k-1} \in \mathbb{R}^{k\times(2n+1)}$ is a full row rank matrix. Clearly, the constraints $Ap = b$ are linear constraints on the coefficients of the polynomial $p(x) = \sum_{i=0}^{2n} p_i x^i$ whereas the constraint $p \in \mathcal{K}_{\mathbb{R}}$ is semi-infinite. Note that the number $k$ of linear constraints must satisfy $1 \leq k \leq 2n+1$. Moreover, if $k = 2n+1$, (11) is clearly not an optimization problem.

From a computational point of view, the problem dual to (11) has a considerable advantage over its primal counterpart. It reads as follows

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } s + \sum_{i=0}^{k-1} a_i y_i = c, \\ \quad s \in \mathcal{K}_{\mathbb{R}}^{\star}. \end{array} \tag{12}$$

Since its constraints are equivalent to $H(c - A^T y) \succeq 0$, the Hankel structure allows us to solve this dual problem efficiently [6].

Using Theorem 1, the primal optimization problem (11) can also be recast as a semidefinite programming problem :

$$\begin{array}{l} \min \langle H(c), Y\rangle \\ \text{s.t. } \langle H(a_i), Y\rangle = b_i,\ i = 0, \ldots, k-1, \\ \quad Y \in \mathcal{S}_+^{n+1}. \end{array}$$

Let us now focus on interpolation constraints. Clearly, an interpolation constraint on a polynomial $p$ is a linear constraint :

$$p(x_i) = \langle p, \pi_{2n}(x_i)\rangle = b_i.$$

Assume that all linear constraints of (11) are interpolation constraints, i.e.

$$\langle a_i, p\rangle \doteq \langle \pi_{2n}(x_i), p\rangle = b_i, \quad i = 0, \ldots, k-1. \tag{13}$$

Then the dual problem (12) is equivalent to

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } H(c) - \sum_{i=0}^{k-1} y_i H(\pi_{2n}(x_i)) \succeq 0. \end{array}$$

As the Hankel structure satisfies

$$H(\pi_{2n}(x)) = \pi_n(x)\pi_n(x)^T, \quad \forall x \in \mathbb{R},$$

we finally obtain the following formulation

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } H(c) - V \operatorname{diag}(y) V^T \succeq 0, \end{array} \tag{14}$$

where the *Vandermonde matrix* $V$ is defined by the nodes $\{x_0, \ldots, x_{k-1}\}$, i.e.

$$V = \begin{bmatrix} 1 & \ldots & 1 \\ x_0 & \ldots & x_{k-1} \\ \vdots & & \vdots \\ x_0^n & \ldots & x_{k-1}^n \end{bmatrix}.$$

**Assumption 1.** *The components of vector b are strictly positive.*

*Remark 3.* Since we work with non-negative polynomials, this assumption is not restrictive. If there exists an integer $i$ such that $b_i = 0$, one can factorize $p(x)$ as $p(x) = \tilde{p}(x)(x - x_i)^2$ and rewrite the optimization problem using the polynomial $\tilde{p}(x)$.

### 3.2 Unit Circle

Several important optimization problems on the unit circle can be formulated as the following *primal* problem

$$\begin{array}{l} \min \langle c, p \rangle_{\mathbb{C}} \\ \text{s.t. } \langle a_i, p \rangle_{\mathbb{C}} = b_i,\ i = 0, \ldots, k-1, \\ \quad p \in \mathcal{K}_{\mathbb{C}}, \end{array} \tag{15}$$

with linearly independent constraints. From a computational point of view, the problem dual to (15) has again a considerable advantage over its primal counterpart. This dual problem reads as follows

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } s + \sum_{i=0}^{k-1} y_i a_i = c, \\ \quad s \in \mathcal{K}_{\mathbb{C}}^{\star}. \end{array} \tag{16}$$

As in the real line setting, one can use the Toeplitz structure of its constraints to get fast algorithms. Using Theorem 2, the primal optimization problem (15) can be reformulated as the semidefinite programming problem

$$\begin{array}{l} \min \langle T(c), Y \rangle \\ \text{s.t. } \langle T(a_i), Y \rangle = b_i,\ i = 0, \ldots, k-1, . \\ \quad Y \in \mathcal{H}_+^{n+1}. \end{array}$$

An interpolation constraint on the trigonometric polynomial $p$ corresponds to

$$p(\theta_i) = \sum_{k=0}^{n} [a_k \cos(k\theta_i) + b_k \sin(k\theta_i)] = b_i \geq 0, \quad \theta_i \in [0, 2\pi],$$

and is equivalent to the linear constraint

$$\langle a_i, p \rangle \doteq p(z_i) = \langle p, \pi_n(z_i) \rangle = b_i, \quad z_i = \mathrm{e}^{\mathrm{i}\theta_i}. \tag{17}$$

Note that the identity

$$T(\pi_{2n}(z)) = \pi_n(z)\pi_n(z)^*, \quad \forall z \in \mathbb{T},$$

holds for the Toeplitz structure. If all linear constraints of (15) are interpolation constraints, the dual can therefore be written as

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s. t. } T(c) - V \operatorname{diag}(y) V^* \succeq 0, \end{array} \tag{18}$$

where the *Vandermonde matrix* $V$ is defined by the points $\{z_0, \ldots, z_{k-1}\}$, i.e.

$$V = \begin{bmatrix} 1 & \ldots & 1 \\ z_0 & \ldots & z_{k-1} \\ \vdots & & \vdots \\ z_0^n & \ldots & z_{k-1}^n \end{bmatrix}.$$

As before, we make the next non-restrictive assumption.

**Assumption 2.** *The components of vector b are strictly positive.*

## 4 Solving the Optimization Problem

In this section, we focus on problems with interpolation conditions, see (13) and (17). We discuss the interpretation of the so-called "strict feasibility" assumption in our context of polynomials. Then we give the explicit solution of three particular optimization problems (one interpolation constraint, two interpolation constraints, property of the objective function). In the general setting, we show that solving the dual problem can be done very efficiently, provided that strict feasibility holds.

### 4.1 Strict Feasibility

The standard assumption on the primal and dual problems is the so-called "strict feasibility" assumption. This assumption is necessary in order to properly define the primal and dual central-paths and thus to solve our pair of primal and dual problems [11]. Moreover, it ensures that the optimal values of both problems coincide, which is an important property to solve our class of problem efficiently.

**Assumption 3 (Strict feasibility).** *There exist points* $\tilde{p} \in \operatorname{int} \mathcal{K}$, $\tilde{s} \in \operatorname{int} \mathcal{K}^\star$ *and* $\tilde{y} \in \mathbb{R}^k$ *that satisfy the following linear systems*

$$\begin{aligned} \langle a_i, \tilde{p} \rangle &= b_i, \quad 0 \leq i \leq k-1, \\ \hat{s} + \sum_{i=0}^{k-1} a_i \tilde{y}_i &= c. \end{aligned}$$

As mentioned in Table 1, the interiors of the primal and dual cones are characterized in terms of polynomials and structured matrices, respectively. However, our particular problem classes allow us to further discuss the interpretation of the previous assumption. More specifically, we shall see that some information about strict feasibility of our problems is known in advance.

**Table 1.** Interiors of primal and dual cones

| | $\mathcal{K} = \mathcal{K}_{\mathbb{R}}$ | $\mathcal{K} = \mathcal{K}_{\mathbb{C}}$ |
|---|---|---|
| $p \in \operatorname{int} \mathcal{K}$ | $p(t) > 0, \forall t \in \mathbb{R}$ and $p_{2n} > 0$ | $p(z) > 0, \forall \lvert z \rvert = 1$ |
| $s \in \operatorname{int} \mathcal{K}^\star$ | $H(s)$ is positive definite | $T(s)$ is positive definite |

### Real Line

First, we analyze strict feasibility of the primal constraints. If the number of interpolation points is less or equal to $n+1$, i.e. $k \leq n+1$, it is clear that there exists a strictly positive polynomial $\tilde{p}$ such that $A\tilde{p} = b$. Assume that $k = n+1$ and let $\{l_i(x)\}_{i=0}^n$ be the set of Lagrange polynomials of degree $n$ associated with the interpolation points. By definition, these polynomials satisfy the identities

$$l_i(x_j) = \delta_{ij}, \quad 0 \leq i, j \leq n,$$

where $\delta_{ij}$ is the well-known Kronecker delta. The polynomial

$$\tilde{p}(x) = \sum_{i=0}^{n} b_i (l_i(x))^2$$

clearly satisfies all our interpolation constraints and belongs to $\operatorname{int} \mathcal{K}_{\mathbb{R}}$. For the case $k < n+1$, we can add $n+1-k$ "extra" interpolation constraints and check that the (original) primal problem is always strictly feasible. If the number of interpolation points is strictly greater that $n+1$, we cannot say anything in advance about primal strict feasibility.

Let us now analyze strict feasibility of the dual constraints. Because of the structure of our interpolation constraints, the interior of the dual space is the set of vectors $s = c - A^T y$ such that

$$H(s) = H(c - A^T y) = H(c) - \sum_{i=0}^{k-1} y_i \pi_n(x_i)\pi_n(x_i)^T \succ 0.$$

If $k \geq n+1$, we conclude from this inequality that there always exists $s = c - A^T y \in \operatorname{int} \mathcal{K}_{\mathbb{R}}^{\star}$. Another simple situation arises when $c \in \operatorname{int} \mathcal{K}_{\mathbb{R}}^{\star}$, i.e $H(c) \succ 0$. Then the dual problem is always strictly feasible. For instance, this situation occurs when minimizing the integral of the polynomial $p(x)$ on a finite interval $I \subseteq (-\infty, +\infty)$ :

$$\langle c, p \rangle = \int_I p(x)dx = \sum_{i=0}^{2n} p_i (\int_I x^i dx),$$

subject to interpolation constraints. This situation is frequent in practice and one easily checks that $c \in \operatorname{int} \mathcal{K}_{\mathbb{R}}^{\star}$ in this case. Indeed, the inner product $\langle c, p \rangle$ is positive for all non-zero $p \in \mathcal{K}_{\mathbb{R}}$.

*Remark 4.* If the dual problem is strictly feasible, one can always reformulate the problem in order to ensure that $c \in \operatorname{int} \mathcal{K}_{\mathbb{R}}^{\star}$.

We have summarized our discussion on Figure 1. Let us point out a remarkable property of our class of problems, which is clearly exhibited on this figure. If the number of constraints is equal to $n+1$, both primal and dual problems are strictly feasible and this property is *independent* of the data. Except for this particular case, there usually exists a trade-off between strict primal and dual feasibility.

### Unit Circle

Using exactly the same argument, one can show that the primal problem is always strictly feasible if the number of interpolation constraints is less or equal to $n+1$. As in the real line, there exists a trade-off between strict

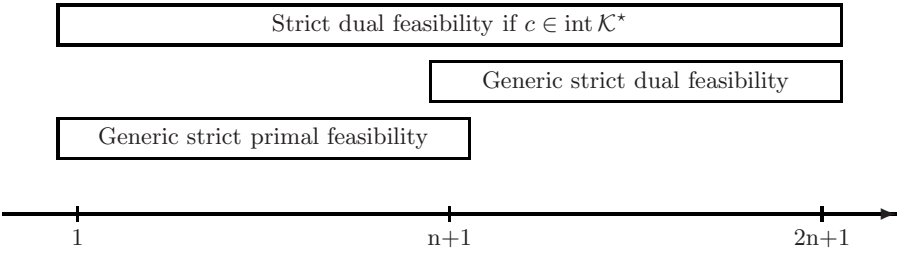



**Fig. 1.** Generic strict feasibility as a function of the number of interpolation constraints

feasibility of the primal and dual constraints unless $k = n + 1$. If $c \in \operatorname{int} \mathcal{K}_{\mathbb{C}}^{\star}$, i.e $T(c) \succ 0$, the dual problem is always strictly feasible.

Therefore, the largest class of interpolation problems on non-negative polynomials (degree $2n$ or $n$, in the real line or unit circle setting, respectively) for which strict feasibility holds and does not depend on the interpolation points, satisfies the following assumption.

**Assumption 4.** *The number $k$ of interpolation constraints is less or equal to $n + 1$ and the objective vector $c$ satisfies $H(c) \succ 0$ (real line setting) or $T(c) \succ 0$ (unit circle setting), i.e. $c \in \operatorname{int} \mathcal{K}^{\star}$.*

From now on, we focus on problems that fulfil this assumption. First, we consider several problems for which explicit solutions are easily computed from the data.

### 4.2 One Interpolation Constraint

#### Real Line

Suppose that one wants to solve the primal problem

$$\min\{\langle c, p\rangle : p(\bar{x}) = b, p \in \mathcal{K}_{\mathbb{R}}\}.$$

The dual problem reads as follows

$$\begin{array}{l} \max\ by \\ \text{s.t. } H(c) \succeq y\pi_n(\bar{x})[\pi_n(\bar{x})]^T. \end{array}$$

Without loss of generality, the scalar $b$ is assumed to be equal to 1. Therefore, the optimal value of this problem

$$\frac{1}{\langle H(c)^{-1}\pi_n(\bar{x}), \pi_n(\bar{x})\rangle},$$

is equal to the optimal value of $y$. Using Assumption 4, the optimal vector $p$ is thus given by

$$p = H^*(qq^T), \quad q = \frac{H(c)^{-1}\pi_n(\bar{x})}{\langle H(c)^{-1}\pi_n(\bar{x}), \pi_n(\bar{x})\rangle}.$$

One can check that

$$\begin{aligned} p(\bar{x}) = \langle \pi_{2n}(\bar{x}), p\rangle = \langle \pi_n(\bar{x})\pi_n(\bar{x}), qq^T\rangle = (\langle \pi_n(\bar{x}), q\rangle)^2 = 1, \\ \langle c, p\rangle = \langle H(c), qq^T\rangle = \frac{1}{\langle H(c)^{-1}\pi_n(\bar{x}), \pi_n(\bar{x})\rangle}. \end{aligned}$$

As $p$ is feasible and the corresponding objective value $\langle c, p\rangle$ is equal to the dual optimal one, the polynomial $p(x) = \langle p, \pi_{2n}(x)\rangle$ is optimal.

**Unit Circle**

Let us now solve the primal problem

$$\min\{\langle c, p\rangle : p(\bar{z}) = \langle p, \pi_n(\bar{z})\rangle_{\mathbb{R}} = b, p \in \mathcal{K}_{\mathbb{C}}\}. \tag{19}$$

As in the real line setting, both primal and dual optimal solutions are computed explicitly by making use of Assumption 4. They are equal to :

$$y = \frac{1}{\langle T(c)^{-1}\pi_n(\bar{z}), \pi_n(\bar{z})\rangle},$$
$$p = T^*(qq^*), \quad q = \frac{T(c)^{-1}\pi_n(\bar{z})}{\langle T(c)^{-1}\pi_n(\bar{z}), \pi_n(\bar{z})\rangle}.$$

*Example 1 (Moving average system, [14]).* Let $h[n]$ be a discrete time signal and $\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})$ be its Fourier transform. The function $|\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})|^2$ is known as the *energy density spectrum* because it determines how the energy is distributed in frequency. Let us compute the signal that has the minimum energy

$$2\pi E = \int_{-\pi}^{\pi} |\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})|^2 d\omega$$

and satisfies $|\mathcal{H}(\mathrm{e}^{\mathrm{i}0})| = 1$.

This is exactly an example of the problem class (19). Since $p(\mathrm{e}^{\mathrm{i}\omega}) = |\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})|^2$ is a trigonometric polynomial, $\int_{-\pi}^{\pi} p(\mathrm{e}^{\mathrm{i}\omega})d\omega = p_0$. The vector $c$ that defines the objective function is thus equal to $c = [1, 0, \ldots, 0]^T = e_0$. The interpolation constraint is obviously defined by $\bar{z} = \pi_n(\mathrm{e}^{\mathrm{i}0}) = e$ and $b = 1$.

Therefore, the optimal primal solution is given by

$$p = T^*(qq^*), \quad q = \frac{[1, \ldots, 1]^T}{n+1}.$$

and the corresponding Fourier transform $\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})$ can be set to

$$\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega}) = \sum_{i=0}^{n} \frac{1}{n+1}\mathrm{e}^{-\mathrm{i}\omega}.$$

Note that $|\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})|^2$ is an approximation of a low-pass filter, see Figure 2. The corresponding signal is exactly the impulse response of the *moving average system* :

$$h[k] = \begin{cases} \frac{1}{n+1}, & 0 \le k \le n+1, \\ 0, & \text{otherwise.} \end{cases}$$

Since convolution of a discrete signal $x[n]$ with $h[n]$ returns a signal $y[n]$ such that

$$y[k] = \frac{1}{n+1}\sum_{l=0}^{n} x[k-l],$$

$y[n]$ is the "moving average" of $x[n]$.

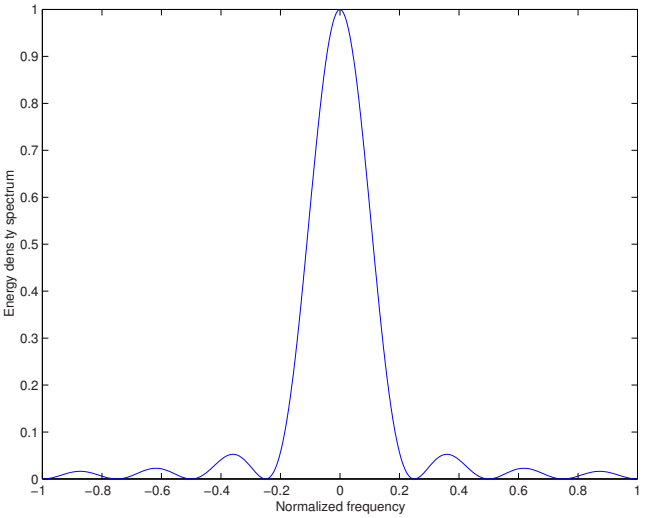



**Fig. 2.** Energy density spectrum ($|\mathcal{H}(\mathrm{e}^{\mathrm{i}\omega})|^2$ – n=7)

### 4.3 Two Interpolation Constraints

Before investigating problems with two interpolation constraints, we need to solve explicitly a 2-dimensional semidefinite programming problem.

**Proposition 1.** *Let* $b_0, b_1 \in \operatorname{int} \mathbb{R}_+$ *and* $\alpha, \gamma \in \mathbb{R}$ *and* $\beta \in \mathbb{C}$. *The optimal value of the optimization problem*

$$\begin{aligned} &\max\ b_0 y_0 + b_1 y_1 \\ &\text{s. t. } \begin{bmatrix} \alpha & \overline{\beta} \\ \beta & \gamma \end{bmatrix} \succeq \begin{bmatrix} y_0 & 0 \\ 0 & y_1 \end{bmatrix} \end{aligned} \tag{20}$$

*is reached at the optimal point*

$$y_0 = \alpha - |\beta| \sqrt{\frac{b_1}{b_0}}, \qquad y_1 = \gamma - |\beta| \sqrt{\frac{b_0}{b_1}}$$

*and it is equal to* $b_0\alpha + b_1\gamma - 2|\beta|\sqrt{b_0 b_1}$.

*Proof.* The constraints are equivalent to

$$\alpha - y_0 \geq 0, \qquad \gamma - y_1 - \frac{|\beta|^2}{\alpha - y_0} \geq 0.$$

Maximizing the linear function $b_0 y_0 + b_1 y_1$ on this 2-dimensional convex region is straightforward (see Figure 3). Clearly, the system of equations

$$\frac{|\beta|^2}{(\alpha - y_0)^2} = \frac{b_0}{b_1}, \quad y_1 = \gamma - \frac{|\beta|^2}{\alpha - y_0},$$

provides us with the optimal point $(y_0, y_1)$. □

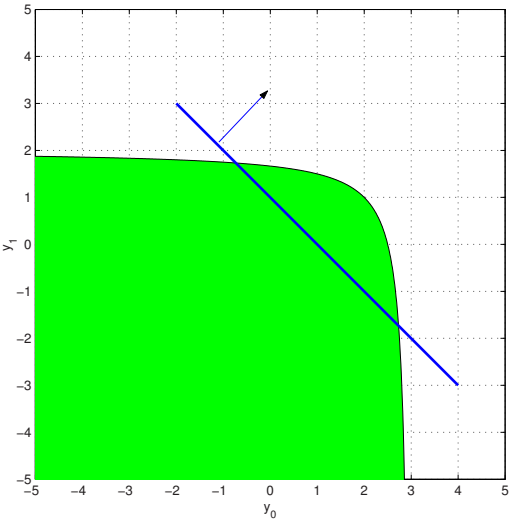



**Fig. 3.** Feasibility region of (20) with $\alpha = 3, \beta = (1+\mathrm{i})/\sqrt{2}$ and $\gamma = 2$

### Real Line

If the number of interpolation constraints is equal to 2, the dual problem (12) is given by

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } H(c) \succeq y_0 \pi_n(x_0)[\pi_n(x_0)]^T + y_1 \pi_n(x_1)[\pi_n(x_1)]^T \end{array} .$$

Equivalently, the dual constraint can be recast as

$$H(c) - \begin{bmatrix} \pi_n(x_0) & \pi_n(x_1) \end{bmatrix} \begin{bmatrix} y_0 & 0 \\ 0 & y_1 \end{bmatrix} \begin{bmatrix} \pi_n(x_0) & \pi_n(x_1) \end{bmatrix}^T \succeq 0.$$

Let us define the matrix $M_H(c; x_0, x_1)$ by

$$M_H(c; x_0, x_1) = \begin{bmatrix} \langle H(c)^{-1}\pi_n(x_0), \pi_n(x_0)\rangle & \langle H(c)^{-1}\pi_n(x_1), \pi_n(x_0)\rangle \\ \langle H(c)^{-1}\pi_n(x_0), \pi_n(x_1)\rangle & \langle H(c)^{-1}\pi_n(x_1), \pi_n(x_1)\rangle \end{bmatrix}.$$

If $\mathrm{diag}(y)$ is positive definite at the optimum, then the previous linear matrix inequality can be recast as

$$M_H(c; x_0, x_1)^{-1} \succeq \mathrm{diag}(y).$$

Indeed, this reformulation follows from the Schur complement formula. Otherwise, our hypothesis on the objective function, $c \in \mathrm{int}\, \mathcal{K}_{\mathbb{R}}^{\star}$, can be used so as to obtain the same reformulation. We delay the proof of this fact to the general setting, see Proposition 3.

Consequently, Proposition 1 allows us to solve our dual problem explicitly :

$$y_0 = \frac{1}{\det(M_H)} \left[ \langle H(c)^{-1}\pi_n(x_1), \pi_n(x_1)\rangle - |\langle H(c)^{-1}\pi_n(x_0), \pi_n(x_1)\rangle| \sqrt{\frac{b_1}{b_0}} \right],$$

$$y_1 = \frac{1}{\det(M_H)} \left[ \langle H(c)^{-1}\pi_n(x_0), \pi_n(x_0)\rangle - |\langle H(c)^{-1}\pi_n(x_0), \pi_n(x_1)\rangle| \sqrt{\frac{b_0}{b_1}} \right],$$

with $\det(M_H) = \det(M_H(c; x_0, x_1))$.

Our primal optimization problem can also be solved explicitly. To see this, define the vector $v = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$ as the solution of the linear system

$$\begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} M_H(c; x_0, x_1) \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} \sqrt{b_0} \\ \sqrt{b_1} \end{bmatrix}$$

where $\sigma \in \{-1, +1\}$ is the sign of $\langle H(c)^{-1}\pi_n(x_0), \pi_n(x_1)\rangle$. Then the vector

$$p = H^*(qq^*), \quad q = H(c)^{-1} \begin{bmatrix} \pi_n(x_0) & \pi_n(x_1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$$

defines a non-negative polynomial $p(x) = (\langle q, \pi_n(x)\rangle)^2$ that satisfies $p(x_0) = b_0$ and $p(x_1) = b_1$. Indeed, we have

$$\begin{bmatrix} q(x_0) \\ q(x_1) \end{bmatrix} = \begin{bmatrix} \pi_n(x_0) & \pi_n(x_1) \end{bmatrix}^T q = \begin{bmatrix} \sqrt{b_0} \\ \sigma\sqrt{b_1} \end{bmatrix}$$

Moreover, the inner product $\langle c, p\rangle$ is equal to the optimal dual value : the vector $p$ is thus optimal.

### Unit Circle

As in the real line setting, the dual problem can be rewritten as

$$\begin{aligned} &\max \langle b, y\rangle \\ &\text{s. t. } M_T(c; z_0, z_1)^{-1} \succeq \operatorname{diag}(y) \end{aligned}$$

where

$$M_T(c; z_0, z_1) = \begin{bmatrix} \langle T(c)^{-1}\pi_n(z_0), \pi_n(z_0)\rangle & \langle T(c)^{-1}\pi_n(z_1), \pi_n(z_0)\rangle \\ \langle T(c)^{-1}\pi_n(z_0), \pi_n(z_1)\rangle & \langle T(c)^{-1}\pi_n(z_1), \pi_n(z_1)\rangle \end{bmatrix}.$$

The optimal dual solution is now equal to

$$y_0 = \frac{1}{\det(M_T)} \left[ \langle T(c)^{-1}\pi_n(z_1), \pi_n(z_1)\rangle - |\langle T(c)^{-1}\pi_n(z_0), \pi_n(z_1)\rangle| \sqrt{\frac{b_1}{b_0}} \right],$$

$$y_1 = \frac{1}{\det(M_T)} \left[ \langle T(c)^{-1}\pi_n(z_0), \pi_n(z_0)\rangle - |\langle T(c)^{-1}\pi_n(z_0), \pi_n(z_1)\rangle| \sqrt{\frac{b_0}{b_1}} \right],$$

with $\det(M_T) = \det(M_T(c; z_0, z_1))$. Let us define the vector $\begin{bmatrix} v_0 & v_1 \end{bmatrix}^T$ as the solution of the linear system

$$\begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix}^* M_T(c; z_0, z_1) \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} \sqrt{b_0} \\ \sqrt{b_1} \end{bmatrix}$$

where $\sigma$ is equal to $\mathrm{e}^{-\mathrm{i}\arg\langle T(c)^{-1}\pi_n(z_1), \pi_n(z_0)\rangle}$. The vector

$$p = T^*(qq^*), \quad q = T(c)^{-1} \begin{bmatrix} \pi_n(z_0) & \pi_n(z_1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$$

corresponds to a trigonometric polynomial $p(z) = |q(z)|^2$ that satisfies our interpolation constraints and such that $\langle c, p\rangle = \langle b, y\rangle$. This vector $p$ is thus the (primal) optimal one.

### 4.4 More Interpolation Constraints ($k \leq n+1$)

If Assumption 4 holds and $k \leq n+1$, the previous analysis can always be carried out. We first focus on the unit circle setting and show the connection with spectral factorization of trigonometric polynomials. The real line problem is then solved using a similar methodology. Let us start with two preliminary results.

#### Preliminary Results

**Proposition 2.** *Let $C \in \operatorname{int} \mathcal{H}_+^n$ be a positive definite matrix and $V = \begin{bmatrix} V_0 & V_1 \end{bmatrix} \in \mathbb{C}^{n\times n}$ be a nonsingular matrix. If the matrix $W = \begin{bmatrix} W_0 \\ W_1 \end{bmatrix}$ is the (left) inverse of $V$ with compatible partitions, i.e. $\begin{bmatrix} W_0V_0 & W_0V_1 \\ W_1V_0 & W_1V_1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$, then we have*

$$(V_1^*C^{-1}V_1)^{-1} = W_1CW_1^* - W_1CW_0^*(W_0CW_0^*)^{-1}W_0CW_1.$$

*Proof.* Let us apply the well-known *Schur complement identity*

$$\begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} I & 0 \\ G\mathrm{e}^{-1} & I \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & H - G\mathrm{e}^{-1}F \end{bmatrix} \begin{bmatrix} I & \mathrm{e}^{-1}F \\ 0 & I \end{bmatrix}, \quad \det(E) \neq 0$$

to the matrix product

$$WCW^* = \begin{bmatrix} W_0CW_0^* & W_0CW_1^* \\ W_1CW_0^* & W_1CW_1^* \end{bmatrix}.$$

Clearly, we obtain that

$$WCW^* = \begin{bmatrix} I & 0 \\ (W_1CW_0^*)(W_0CW_0^*)^{-1} & I \end{bmatrix} \begin{bmatrix} W_0CW_0^* & 0 \\ 0 & W_1/W_0 \end{bmatrix} \begin{bmatrix} I & (W_0CW_0^*)^{-1}(W_0CW_1^*) \\ 0 & I \end{bmatrix}$$

with $W_1/W_0 = W_1CW_1^* - (W_1CW_0^*)(W_0CW_0^*)^{-1}(W_0CW_1^*)$. Because the matrix $WCW^*$ is nonsingular (by assumption), we have

$$(WCW^*)^{-1} = \begin{bmatrix} I & -(W_0CW_0^*)^{-1}(W_0CW_1^*) \\ 0 & I \end{bmatrix} \begin{bmatrix} W_0CW_0^* & 0 \\ 0 & W_1/W_0 \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ -(W_1CW_0^*)(W_0CW_0^*)^{-1} & I \end{bmatrix}.$$

Hence, the lower right block of the identity

$$(WCW^*)^{-1} = V^*C^{-1}V = \begin{bmatrix} V_0^*C^{-1}V_0 & V_0^*C^{-1}V_1 \\ V_1^*C^{-1}V_0 & V_1^*C^{-1}V_1 \end{bmatrix}$$

is exactly equivalent to

$$V_1^*C^{-1}V_1^* = (W_1CW_1^* - (W_1CW_0^*)(W_0CW_0^*)^{-1}(W_0CW_1^*))^{-1}.$$

□

**Proposition 3.** *Let $C \in \operatorname{int} \mathcal{H}_+^n$ be a positive definite matrix and $V_1 \in \mathbb{C}^{n\times k}$ be a matrix with full column rank ($k \leq n$). Then the linear matrix inequality*

$$C \succeq V_1 \operatorname{diag}(y) V_1^* \tag{21}$$

*is equivalent to*

$$(V_1^*C^{-1}V_1)^{-1} \succeq \operatorname{diag}(y). \tag{22}$$

*Proof.* If $k = n$, the proof is trivial. Indeed, both inequalities (21) and (22) are congruent. This congruence is defined by the nonsingular matrix $V_1^{-1}$. If $k < n$, Proposition 2 must be used. Let $V_0 \in \mathbb{C}^{n\times(n-k)}$ be a matrix such that $V = \begin{bmatrix} V_0 & V_1 \end{bmatrix} \in \mathbb{C}^{n\times n}$ is nonsingular. The (left) inverse of $V$ is denoted by $W = \begin{bmatrix} W_0 \\ W_1 \end{bmatrix}$. If the rows of $W$ are partitioned according to the partition of $V$, we have

$$WV = \begin{bmatrix} W_0V_0 & W_0V_1 \\ W_1V_0 & W_1V_1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

The linear matrix inequality (21), which can be rewritten as

$$C - \begin{bmatrix} V_0 & V_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \operatorname{diag}(y) \end{bmatrix} \begin{bmatrix} V_0^* \\ V_1^* \end{bmatrix} \succeq 0,$$

is thus equivalent to

$$\begin{bmatrix} W_0 \\ W_1 \end{bmatrix} C \begin{bmatrix} W_0^* & W_1^* \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \operatorname{diag}(y) \end{bmatrix} \succeq 0 \tag{23}$$

by congruence. Because $W_0 C W_0^*$ is positive definite by assumption, the previous inequality is equivalent to positive semidefiniteness of its Schur complement in (23),

$$W_1 C W_1^* - (W_1 C W_0^*)(W_0 C W_0^*)^{-1}(W_0 C W_1^*) \succeq \operatorname{diag}(y).$$

We complete the proof by making use of Proposition 2. □

**Unit Circle**

Remember that the optimization problem of interest is

$$\begin{aligned} &\min \langle c, p \rangle_{\mathbb{R}} \\ &\text{s. t. } \langle p, \pi_n(z_i) \rangle_{\mathbb{R}} = b_i,\ i = 0, \ldots, k-1, \\ &\qquad p \in \mathcal{K}_{\mathbb{C}}. \end{aligned} \tag{24}$$

If the non-negative trigonometric polynomial $p(z)$ is written as a square by making use of an arbitrary spectral factor $q(z)$, i.e. $p(z) = |q(z)|^2$ or $p = T^*(qq^*)$, the primal optimization problem can be rewritten as

$$\begin{aligned} &\min \langle T(c)q, q \rangle \\ &\text{s. t. } \langle q, \pi_n(z_i) \rangle = \sqrt{b_i} \mathrm{e}^{\mathrm{i}\theta_i},\ i = 0, \ldots, k-1, \end{aligned} \tag{25}$$

where $\{\theta_i\}_{i=0}^{k-1}$ is a set of phases.

Define the vector $\sigma$ by $\sigma_i = \sqrt{b_i} \mathrm{e}^{\mathrm{i}\theta_i}, i = 0, \ldots, k-1$ and the matrix $M_T$ by

$$M_T(c; z_0, \ldots, z_{k-1}) = \left[\pi_n(z_0) \cdots \pi_n(z_{k-1})\right]^* T(c)^{-1} \left[\pi_n(z_0) \cdots \pi_n(z_{k-1})\right].$$

As a function of $\sigma$, the optimal solution of (25) is equal to

$$q = T(c)^{-1} \left[\pi_n(z_0) \cdots \pi_n(z_{k-1})\right] M_T(c; z_0, \ldots, z_{k-1})^{-1} \sigma \tag{26}$$

and the corresponding optimal value is

$$\langle T(c)q, q \rangle = \langle M_T(c; z_0, \ldots, z_{k-1})^{-1} \sigma, \sigma \rangle.$$

*Remark 5.* A direct consequence of (26) is that our spectral factor $q(z)$ is decomposed as a sum of "Lagrange-like" polynomials :

$$q(z) = \langle q, \pi_n(z) \rangle_{\mathbb{C}} = \sum_{i=0}^{k-1} \mathrm{e}^{\mathrm{i}\theta_i} \sigma_i l_i(z)$$

where $l_i(z_j) = \delta_{ij}, \forall i, j$.

Finally, the optimal solution of problem (25) is obtained by minimizing over the vector $\sigma$,

$$\begin{array}{l} \min \langle M_T(c; z_0, \ldots, z_{k-1})^{-1}\sigma, \sigma\rangle \\ \text{s.t. } |\sigma_i|^2 = b_i, \quad i = 0, \ldots, k-1. \end{array} \tag{27}$$

If $m > 2$, an explicit solution is difficult to obtain easily from this new formulation. However, we can solve the semidefinite relaxation of problem (27) :

$$\begin{array}{l} \min \langle M_T^{-1}(z_0, \ldots, z_{k-1}), X\rangle, \\ \text{s.t. } \operatorname{diag}(X) = b, \\ \qquad X \in \mathcal{H}_+^k, \end{array} \tag{28}$$

where $\operatorname{diag}(X)$ is the vector defined by the diagonal elements of $X$. In general, a quadratic problem of the form (27) is NP-hard to solve, see the Appendix. Nevertheless, the particular structure of the quadratic objective function yields an extremely interesting result.

**Theorem 3.** *If Assumption 4 holds, relaxation (28) of quadratically constrained quadratic problem (27) is exact.*

*Proof.* Using standard convex duality theory, the dual of problem (28) is

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } M_T^{-1}(z_0, \ldots, z_{k-1}) \succeq \operatorname{diag}(y), \end{array} \tag{29}$$

which is exactly the dual of the original problem (24) :

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } T(c) \succeq \left[\pi_n(z_0) \ldots \pi_n(z_{k-1})\right] \operatorname{diag}(y) \left[\pi_n(z_0) \ldots \pi_n(z_{k-1})\right]^* . \end{array} \tag{30}$$

To see this, we define the matrix $V_1$ as $V_1 = \left[\pi_n(z_0) \ldots \pi_n(z_{k-1})\right]$ and we apply Proposition 3 with $C = T(c)$. Because the (dual) constraints of (29) and (30) are equivalent, both problems are identical.

By assumption the original problem (24) has no duality gap. Since both problems (24) and (28) have the same dual, the relaxation has also a zero duality gap. This last observation completes our proof. □

The optimal coefficients $p$ can be obtained from the solution $X$ of (28) via the identity

$$p = T^*(T(c)^{-1}V_1 M_T^{-1} X M_T^{-1} V_1^* T(c)^{-1})$$

where $V_1 = \left[\pi_n(z_0) \ldots \pi_n(z_{k-1})\right]$ and $M_T = M_T(c; z_0, \ldots, z_{k-1})$. To see this, note that

$$\begin{aligned}\langle c, p\rangle &= \langle T(c), T(c)^{-1}V_1 M_T^{-1} X M_T^{-1} V_1^* T(c)^{-1}\rangle \\ &= \langle T(c)^{-1}V_1 M_T^{-1} X M_T^{-1} V_1^*, I\rangle \\ &= \langle V_1^* T(c)^{-1}V_1 M_T^{-1} X M_T^{-1}, I\rangle \\ &= \langle X, M_T^{-1}\rangle\end{aligned}$$

and that, for all $i$,

$$\begin{aligned}\langle p, \pi_n(z_i)\rangle =&\langle T(\pi_n(z_i)), T(c)^{-1}V_1 M_T^{-1} X M_T^{-1} V_1^* T(c)^{-1}\rangle \\ =&\langle \pi_n(z_i)\pi_n(z_i)^*, T(c)^{-1}V_1 M_T^{-1} X M_T^{-1} V_1^* T(c)^{-1}\rangle \\ =&\langle (\pi_n(z_i)^* T(c)^{-1}V_1 M_T^{-1}) X (M_T^{-1} V_1^* T(c)^{-1}\pi_n(z_i)), I\rangle \\ =&\langle e_i e_i^*, X\rangle = b_i.\end{aligned}$$

**Real Line**

Remember that the optimization problem of interest is

$$\begin{aligned}&\min \langle c, p\rangle \\ &\text{s.t. } \langle p, \pi_{2n}(x_i)\rangle = b_i,\ i = 0, \ldots, k-1,. \\ &\qquad p \in \mathcal{K}_{\mathbb{R}}.\end{aligned} \tag{31}$$

If we use any complex spectral factor $q(x)$ of our unknown polynomial $p(x) = |q(x)|^2$ as a variable, the previous analysis can be carried out in the real line setting. It leads exactly to the same formulae *provided that* the following substitutions are performed :

1. $T(c)$ is replaced by $H(c)$;
2. the interpolation points $\{z_i\}_{i=0}^{k-1}$ are replaced by $\{x_i\}_{i=0}^{k-1}$;
3. the matrix $M_T(c; z_0, \ldots, z_{k-1})$ is replaced by its "Hankel counterpart"

$$[M_H(c; x_0, \ldots, x_{k-1})]_{ij} = \pi_n(x_i)^* H(c)^{-1} \pi_n(x_j).$$

Let us summarize the most important steps. First, the primal optimization problem (31) is reformulated as

$$\begin{aligned}&\min \langle H(c)q, q\rangle \\ &\text{s.t. } \langle q, \pi_n(x_i)\rangle = \sqrt{b_i}\mathrm{e}^{\mathrm{i}\theta_i},\ i = 0, \ldots, k-1,\end{aligned} \tag{32}$$

which is equivalent to

$$\begin{aligned}&\min \langle M_H(c; x_0, \ldots, x_{k-1})^{-1}\sigma, \sigma\rangle \\ &\text{s.t. } |\sigma_i|^2 = b_i, \quad i = 0, \ldots, k-1.\end{aligned} \tag{33}$$

In practice, this last optimization problem is solved using the following relaxation

$$\begin{aligned}&\min \langle M_H^{-1}(x_0, \ldots, x_{k-1}), X\rangle, \\ &\text{s.t. } \mathrm{diag}(X) = b, \\ &\qquad X \in \mathcal{H}_+^k.\end{aligned} \tag{34}$$

As before, the structure of quadratic problem (33) leads to an exact semidefinite relaxation.

**Theorem 4.** *If Assumption 4 holds, relaxation (34) of quadratically constrained quadratic problem (33) is exact.*

*Proof.* Using standard convex duality theory, the dual of problem (28) is

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } M_H^{-1}(x_0, \ldots, x_{k-1}) \succeq \operatorname{diag}(y), \end{array} \tag{35}$$

which is exactly the dual of the original problem (31) :

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } H(c) \succeq \left[\pi_n(x_0) \ldots \pi_n(x_{k-1})\right] \operatorname{diag}(y) \left[\pi_n(x_0) \ldots \pi_n(x_{k-1})\right]^* . \end{array} \tag{36}$$

To see this, we define the matrix $V_1$ as $V_1 = \left[\pi_n(z_0) \ldots \pi_n(z_{k-1})\right]$ and we apply Proposition 3 with $C = T(c)$ and $V_1$. Because the (dual) constraints of (35) and (36) are equivalent, both problems are identical.

By assumption the original problem (31) has no duality gap. Since both problems (31) and (34) have the same dual, the relaxation has also a zero duality gap. This last observation completes our proof. □

**Complexity**

The complexity of solving relaxation (34) or (28) is only a function of the desired accuracy $\epsilon$ and the number of interpolation constraints $k$. If Assumption 4 holds and if the original problem has been pre-processed, it can be solved in a number of iterations that does *not* depend on the degree $n$. Indeed, solving the dual problem (35) or (29) using a *standard* path-following scheme requires $\mathcal{O}(\sqrt{k} \log \frac{1}{\epsilon})$ Newton steps. At each iteration, computing the gradient and the Hessian of a barrier function of the type

$$f(y) = -\log \det(M^{-1} - \operatorname{diag}(y))$$

requires $\mathcal{O}(k^3)$ flops. Note that the pre-processing can be done via fast Hankel or Toeplitz solvers, see [9].

### 4.5 Still More Interpolation Constraints ($m > n + 1$)

If the number of interpolation constraints is strictly greater than $n + 1$, strict feasibility of the primal problem depends on the data. Therefore, a *general* procedure that solves efficiently the primal problem and uses the structure of the interpolation constraints is not likely to exist. Indeed, the primal problem might be infeasible ! Let us illustrate this fact by a simple example.

*Example 2.* Consider the set of polynomials of degree $2n = 4$, non-negative on the real line, and four interpolation points $x = [-2, -1, 1, 2]$. The vector

$b = [1, 1, 1, 1]$ gives a strictly feasible primal problem. Indeed, the polynomial $p(x) = \frac{1}{3}(x^4 - 5x^2 + 7)$ satisfies our interpolation constraints and belongs to $\operatorname{int} \mathcal{K}_{\mathbb{R}}$. If the vector $b$ is equal to $[1, 10, 1, 1]$, the polynomial family that satisfies our interpolation constraints is $p(x; \lambda) = \frac{1}{4}((\lambda - 7)x^4 + 6x^3 + (29 - 5\lambda)x^2 - 24x + 4\lambda)$, $\lambda \in \mathbb{R}$. If $p(x; \lambda)$ belonged to $\operatorname{int} \mathcal{K}_{\mathbb{R}}$, $\lambda$ would be greater than 7. As $p(\frac{5}{4}; \lambda) = \frac{1}{1024}(-371\lambda - 3255), \forall \lambda > 0$, these data correspond to an infeasible primal set...

Of course, the dual structure can still be exploited to try reducing the computational cost. For instance, consider a problem on the unit circle with $m > n+1$ interpolation constraints. Clearly, the corresponding Vandermonde matrix $V$ can be divided into a nonsingular square Vandermonde matrix $V_0$ and a rectangular one $V_1$

$$V = \begin{bmatrix} V_0 & V_1 \end{bmatrix}, \quad \det V_0 \neq 0.$$

If the dual vector is divided accordingly, the dual constraint can be recast as $T(c) \succeq V_0 \operatorname{diag}(y_0) V_0^* + V_1 \operatorname{diag}(y_1) V_1^*$. Since $V_0$ is nonsingular, it is equivalent to

$$V_0^{-1} T(c) V_0^{-*} - V_0^{-1} V_1 \operatorname{diag}(y_1) V_1^*) V_0^{-*} \succeq \operatorname{diag}(y_0).$$

Therefore, an appropriate preprocessing leads to the following dual constraint

$$\hat{C} - \hat{V} \operatorname{diag}(y_1) \hat{V}^* \succeq \operatorname{diag}(y_0).$$

Since the Toeplitz structure of the dual is lost, the resulting algorithm cannot use the underlying displacement operator nor a divide-and-conquer strategy to evaluate the gradient and the Hessian of the self-concordant barrier function. This strategy will thus be slower than the one designed in [6]

### 4.6 Property of the Objective Function

If $H(c)$ or $T(c)$ is not positive definite, the corresponding dual problem can sometimes be solved explicitly.

#### Real Line

If the vector $c$ is such that $H(c)$ can be factorized as

$$H(c) = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} \operatorname{diag}(\lambda_v) & 0 \\ 0 & \operatorname{diag}(\lambda_w) \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix} \tag{37}$$

where $V \in \mathbb{R}^{k \times (n+1)}$ is the Vandermonde matrix defined by the interpolation constraints and $W \in \mathbb{R}^{(n+1-k) \times (n+1)}$ is such that $\begin{bmatrix} V & W \end{bmatrix}$ is full rank, one can easily compute an explicit solution of the optimization. From a theoretical point of view, there exist vectors $c$ such that the proposed factorization does

not exist. From a computational point of view, it may also be difficult to compute accurately.

The dual constraints now reads as follows

$$\begin{bmatrix} \operatorname{diag}(\lambda_v - y) & 0 \\ 0 & \operatorname{diag}(\lambda_w) \end{bmatrix} \succeq 0.$$

If $\operatorname{diag}(\lambda_w)$ is not positive semidefinite, the dual optimization problem is infeasible and the primal problem is unbounded. Otherwise, the solution is obtained by setting the dual variables $y_i$ to their upper bounds, i.e. $y = \lambda_v$. This provides us with either a lower bound or the exact value of the optimization problem, depending on whether the problem has a duality gap.

### Unit Circle

The same factorization technique can be applied to $T(c)$, i.e.

$$T(c) = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} \operatorname{diag}(\lambda_v) & 0 \\ 0 & \operatorname{diag}(\lambda_w) \end{bmatrix} \begin{bmatrix} V^* \\ W^* \end{bmatrix}, \tag{38}$$

and leads to the same results and drawbacks.

## 5 Matrix Polynomials

In this section, we show that most of the previous results still holds in the context of non-negative matrix polynomials. As before, these non-negative polynomials could be defined on the real line, on the imaginary axis and on the unit circle. To avoid redundancies, we only consider the cone of matrix polynomials non-negative on the real line, which is again denoted by $\mathcal{K}_{\mathbb{R}}$,

$$0 \preceq P(x) = \sum_{k=0}^{2n} P_k x^k, \ \forall x \in \mathbb{R}; \qquad P_k = P_k^* \in \mathbb{R}^{q \times q}, \forall k. \tag{39}$$

Theorem 1 can then be extended to the matrix case [6].

**Theorem 5.** *A matrix polynomial $P(x)$ is non-negative on the real axis if and only if there exists a positive semidefinite symmetric block matrix $Y = \{Y_{ij}\}_{i,j=0}^{n}$ such that ($Y_{ij} = 0$ for $i$ or $j$ outside their definition range)*

$$P_k = \sum_{i+j=k} Y_{ij}, \quad \text{for } k = 0, \ldots, 2n. \tag{40}$$

As shown in [6], the dual cone is the set of Hermitian matrix coefficients $S = \begin{bmatrix} S_0 & S_1 & \ldots & S_{2n} \end{bmatrix}$ such that the corresponding block Hankel matrix

$$H(S) = \begin{bmatrix} S_0 & S_1 & \cdots & S_n \\ S_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & S_{2n-1} \\ S_n & \cdots & S_{2n-1} & S_{2n} \end{bmatrix}, \tag{41}$$

is positive semidefinite, i.e. $\mathcal{K}_{\mathbb{R}}^{\star} = \{S : H(S) \succeq 0\}$.

### 5.1 The Optimization Problem

Using matrix interpolation constraints, our optimization problem (11) could be extended to

$$\begin{array}{l} \min \langle C, P\rangle \equiv \sum_{\ell=0}^{2n} \langle C_\ell, P_\ell\rangle \\ \text{s.t. } P(x_i) = \sum_{\ell=0}^{2n} P_\ell x_i^\ell = B_i,\ i = 0, \ldots, k-1\,. \\ \quad\ P(x) \succeq 0,\ \forall x \in \mathbb{R} \end{array} \tag{42}$$

where $\{B_i\}_{i=0}^{k-1}$ is a set of positive definite matrices. Its dual is readily seen to be equal to

$$\begin{array}{l} \max \langle B, Y\rangle \equiv \sum_{i=0}^{k-1} \langle B_i, Y_i\rangle \\ \text{s.t. } S_\ell + \sum_{i=0}^{k-1} x_i^\ell Y_i = C_\ell, \quad \ell = 0, \ldots, 2n\,. \\ \quad\ H(S) \succeq 0, \end{array} \tag{43}$$

### 5.2 Strict Feasibility

As before, primal strict feasibility holds if the number $k$ of matrix interpolation constraints is less or equal to $n+1$. To see this, consider $n+1$ distinct interpolation points $\{x_i\}_{i=0}^{n}$ and the associated Lagrange polynomials $\{L_i(x)\}_{i=0}^{n}$ of degree $n$. These polynomials are defined by the identities

$$L_j(x_i) = \delta_{ij} I_m, \qquad 0 \le i, j \le n.$$

Then the polynomial

$$P(x) = \sum_{i=0}^{n} L_i(x) P(x_i) L_i^T(x) = \sum_{i=0}^{n} L_i(x) B_i L_i^T(x)$$

can be rewritten as

$$P(x) = \langle L\,\mathrm{diag}(\{P(x_i)\}_{i=0}^{k-1}) L^T \Pi_n(x), \Pi_n(x)\rangle$$

where $L$ is nonsingular, $\mathrm{diag}(\{P(x_i)\}_{i=0}^{k-1})$ is positive definite and $\Pi_n(x) = \pi_n(x) \otimes I_m$. By construction, we see that $P(x) \in \mathrm{int}\,\mathcal{K}_{\mathbb{R}}$ and $P(x_i) = B_i, \forall i$.

Since the dual constraints (43) are equivalent to

$$H(C) \succeq \sum_{i=0}^{k-1} \Pi_n(x_i) Y_i \Pi_n(x_i)^T,$$

the dual is strictly feasible if $k \ge n+1$.

Let us state the matrix counterpart of Assumption 4 for future use.

**Assumption 5.** *The number $k$ of interpolation constraints is less or equal to $n+1$ and the objective block vector $C$ satisfies $H(C) \succ 0$.*

Hereafter, we focus on problems satisfying this assumption.

### 5.3 One Interpolation Constraint

Let us consider a matrix interpolation problem with one constraint :

$$\begin{array}{l} \min \langle C, P \rangle \\ \text{s.t. } P(\bar{x}) = \sum_{\ell=0}^{2n} P_\ell \bar{x}^\ell = B \succ 0, \ . \\ \quad P(x) \succeq 0, \ \forall x \in \mathbb{R} \end{array}$$

Without loss of generality, $B$ is assumed to be the identity matrix, i.e. $B = I_m$. Using the dual matrix variable $Y$, the dual problem reads

$$\begin{array}{l} \max \langle I, Y \rangle \\ \text{s.t. } H(C) \succeq \Pi_n(\bar{x}) Y \Pi_n(\bar{x})^T \ . \end{array}$$

Because $H(C) \succ 0$, a standard Schur complement approach shows that the optimal dual variable is

$$Y = [\Pi_n(\bar{x})^T H(C)^{-1} \Pi_n(\bar{x})]^{-1}.$$

The spectral factor

$$Q = H(C)^{-1} \Pi_n(\bar{x}) [\Pi_n(\bar{x})^T H(C)^{-1} \Pi_n(\bar{x})]^{-1}$$

allows us to compute the optimal primal variable $P$

$$P(x) = Q(x)Q(x)^* \quad if and only if \quad P = H^*(QQ^*).$$

It is easy to check that this value of $P$ is optimal, i.e.

$$\langle C, P \rangle = \sum_{\ell=0}^{2n} \langle C_\ell, P_\ell \rangle = \langle H(C)Q, Q \rangle = \langle I, [\Pi_n(\bar{x})^T H(C)^{-1} \Pi_n(\bar{x})]^{-1} \rangle = \langle I, Y \rangle$$

and

$$P(\bar{x}) = \Pi_n(\bar{x})^T Q Q^* \Pi_n(\bar{x}) = I_m = B.$$

### 5.4 More Interpolation Constraints

If the number of matrix interpolation constraints is less or equal to $n+1$, we can again use an arbitrary spectral factor to get an efficient algorithm, the complexity of which mainly depends on $k$ and $m$.

Indeed, let $Q(x)$ be an arbitrary spectral factor $Q(x)$ of our unknown polynomial $P(x)$, i.e $P(x) = Q(x)Q(x)^*$. Then the optimization problem can be rewritten as

$$\begin{array}{l} \min \langle H(C)Q, Q\rangle \\ \text{s. t. } Q(x_i) = \sum_{\ell=0}^{2n} Q_\ell x_i^\ell = B_i^{1/2} U_i,\ i = 0, \ldots, k-1 \end{array} \tag{44}$$

where $\{U_i\}_{i=0}^{k-1}$ is a set of unitary matrices, i.e. $U_i^* U_i = I_m, \forall i$.

If the definition of $M_H$ is adapted to the matrix case,

$$[M_H(C; x_0, \ldots, x_{k-1})]_{ij} = \Pi_n(x_i)^* H(C)^{-1} \Pi_n(x_j),$$

then the optimal solution of (44), written as a function of

$$U = \begin{bmatrix} U_0 \\ \vdots \\ U_{k-1} \end{bmatrix},$$

is equal to

$$Q = H(C)^{-1} \left[\Pi_n(x_0) \cdots \Pi_n(x_{k-1})\right] M_H(C; x_0, \ldots, x_{k-1})^{-1} \operatorname{diag}(\{B_i^{1/2}\}_{i=0}^{k-1}) U.$$

As in the scalar case, the optimal solution of the original problem is obtained via the quadratic optimization problem

$$\begin{array}{l} \min \langle \operatorname{diag}(\{B_i^{1/2}\}_{i=0}^{k-1}) M_H(C; x_0, \ldots, x_{k-1})^{-1} \operatorname{diag}(\{B_i^{1/2}\}_{i=0}^{k-1}) U, U\rangle \\ \text{s. t. } U_i^* U_i = I_m, \quad i = 0, \ldots, k-1. \end{array} \tag{45}$$

The associated semidefinite relaxation is

$$\begin{array}{ll} \min \langle M_H(C; x_0, \ldots, x_{k-1})^{-1}, X\rangle & \\ \text{s. t. } X_{ii} = B_i, & i = 0, \ldots, k-1 \\ \quad X \in \mathcal{H}_+^{mk} & \end{array} \tag{46}$$

where $X_{ii}$ is the $i$th $m \times m$ diagonal block of $X$. Its dual is given by

$$\begin{array}{l} \max \langle B, Y\rangle \\ \text{s. t. } M_H(C; x_0, \ldots, x_{k-1})^{-1} \succeq \operatorname{diag}(\{Y_i\}_{i=0}^{k-1}) \end{array} \tag{47}$$

and is equal to the dual of the original problem. Therefore, we could proceed as before to obtain the following theorem :

**Theorem 6.** *If Assumption 5 holds, relaxation (46) of quadratically constrained quadratic problem (45) is exact.*

Provided that the original problem has been pre-processed, solving the dual problem (47) does *not* depend on the degree $2n$ of $P(x)$. This result is similar to the scalar case. As Assumption 5 guarantees that strict feasibility holds, we obtain an efficient algorithm to solve our problem class.

## 6 Interpolation Conditions on the Derivatives

In this section, we present the straightforward extension of our previous results to interpolation conditions on the derivatives. We only consider the scalar case to keep our equations as small as possible.

### 6.1 Real Line

In the real line setting, interpolation constraints on the derivatives are formulated as

$$p^{(\ell)}(x_i) = \langle p, \pi_{2n}^{(\ell)}(x_i)\rangle = b_i$$

where $\pi_{2n}^{(\ell)}(\cdot)$ is the component-wise $\ell$th derivative of $\pi_{2n}(\cdot)$. Such constraints will be called "interpolation-like" constraints.

If all the linear constraints of (11) are interpolation-like constraints, i.e.

$$\langle a_i, p\rangle \doteq \langle \pi_{2n}^{(\ell_i)}(x_i), p\rangle = b_i, \quad i = 0, \ldots, k-1,$$

the dual problem (12) reads now as follows

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } H(c) - \sum_{i=0}^{k-1} y_i H(\pi_{2n}^{(\ell_i)}(x_i)) \succeq 0. \end{array} \tag{48}$$

Let us now prove that $H(\pi_{2n}^{(\ell_i)}(x_i))$ has a special structure.

**Proposition 4.** *Let $\ell \geq 0$. Then*

$$H(\pi_{2n}^{(\ell)}(x)) = \sum_{r=0}^{\ell} \binom{\ell}{r} \pi_n^{(r)}(x)(\pi_n^{(\ell-r)}(x))^T, \quad \forall x \in \mathbb{R} \tag{49}$$

*and the rank of this matrix is* $\min\{\ell, 2n-\ell\}+1$.

*Proof.* Since $H(\pi_{2n}(x)) = \pi_n(x)\pi_n(x)^T$ and $H(\cdot)$ is a linear operator, equation (49) is a direct consequence of the chain rule. The rank condition originates from the fact that $\pi_n^{(n+1)}(x) = 0$. □

This proposition allows us to improve the formulation (48) of the dual problem. First of all, assume that the interpolation points are distinct and that $\ell_i \leq n, \forall i$. Let us define a block diagonal matrix

$$\Delta(y) = \operatorname{diag}(\{\Delta_0(y), \ldots, \Delta_{k-1}(y)\})$$

where $\Delta_i(y)$ is a $(\ell_i+1)\times(\ell_i+1)$ matrix defined by

$$\Delta_i(y) = \begin{bmatrix} 0 & & \binom{\ell_i}{\ell_i} y_i \\ & \iddots & \\ \binom{\ell_i}{0} y_i & & 0 \end{bmatrix}, \quad i = 0, \ldots, k-1.$$

Using the above proposition, the dual problem can be written as

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } H(c) - V\Delta(y)V^T \succeq 0, \end{array} \tag{50}$$

where $V$ is the non-square *confluent matrix*

$$V = \left[ \pi_n^{(0)}(x_0) \ldots \pi_n^{(\ell_1)}(x_0) \mid \ldots \mid \pi_n^{(0)}(x_m) \ldots \pi_n^{(\ell_m)}(x_m) \right].$$

If the interpolation points are not distinct or if there exists at least one index $i$ such that $\ell_i > n$, the matrix $V$ and the block-diagonal matrix $\Delta(y)$ must be redefined in order to get a dual problem similar to (50). Because the appropriate reformulation is evident, but cumbersome, it has been omitted.

If $H(c) \succ 0$ and the numbers of rows of $V$ is greater than its number of columns, the dual constraint (50) is easily recast using Proposition 3 :

$$(V^T H(c)^{-1} V)^{-1} \succeq \Delta(y).$$

The complexity of solving the dual problem (50) depends mostly on the dimension of $\Delta(y)$. That is, an appropriate preprocessing tends to eliminate the dependence on the degree $2n$. Because primal strict feasibility cannot be guaranteed from the knowledge of $k$, we cannot guarantee that the semidefinite relaxation is exact.

### 6.2 Unit Circle

In the unit circle setting, interpolation constraints on the derivatives, $p^{(\ell_i)}(\theta_i) = b_i$, are equivalent to the linear constraints

$$p^{(\ell_i)}(z_i) = \langle (-\mathrm{i}N)^{\ell_i} p, \pi_n(z_i) \rangle = \langle p, (\mathrm{i}N)^{\ell_i} \pi_n(z_i) \rangle = b_i, \quad z_i = \mathrm{e}^{\mathrm{i}\theta_i} \tag{51}$$

where $N = \mathrm{diag}(0, 1, \ldots, n)$.

If all linear constraints of (15) are interpolation-like constraints, i.e.

$$\langle a_i, p \rangle \doteq \langle p, (\mathrm{i}N)^{\ell_i} \pi_n(z_i) \rangle = b_i, \quad z_i = \mathrm{e}^{\mathrm{i}\theta_i}, \quad i = 0, \ldots, k-1,$$

the dual problem (16) reads now as follows

$$\begin{array}{l} \max \langle b, y \rangle \\ \text{s.t. } T(c) - \sum_{i=0}^{k-1} y_i T((\mathrm{i}N)^{\ell_i} \pi_n(z_i)) \succeq 0 \end{array}. \tag{52}$$

Note that $T((\mathrm{i}N)^{l_i} \pi_n(z))$ has a special structure.

**Proposition 5.** *Let $\ell \geq 0$. Then*

$$T((\mathrm{i}N)^\ell \pi_n(z)) = \sum_{r=0}^{\ell} \binom{\ell}{r} (\mathrm{i}N)^r \pi_n(z)[(\mathrm{i}N)^{\ell-r}\pi_n(z)]^* \tag{53}$$

*and the rank of this matrix is* $\min\{\ell, n\} + 1$.

*Proof.* Since $T(\pi_n(z)) = \pi_n(z)\pi_n(z)^*$, $\frac{\partial}{\partial\theta}(\pi_n(z)|_{z=\mathrm{e}^{\mathrm{i}\theta}}) = \mathrm{i}N(\pi_n(z)|_{z=\mathrm{e}^{\mathrm{i}\theta}})$ and $T(\cdot)$ is a linear operator, it is straightforward to check equation (53). □

Assume that the interpolation points are distinct and define the block diagonal matrix

$$\Delta(y) = \mathrm{diag}(\{\Delta_0(y), \ldots, \Delta_{k-1}(y)\})$$

as before. Using the above proposition, the dual problem can be written as

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } T(c) - W\Delta(y)W^* \succeq 0. \end{array}$$

where $W$ is the non-square matrix

$$W = \Big[(\mathrm{i}N)^0\pi_n^{(0)}(z_0), \ldots, (\mathrm{i}N)^{\ell_1}\pi_n^{(\ell_1)}(z_0)| \ldots | (\mathrm{i}N)^0\pi_n^{(0)}(z_{k-1}), \ldots, (\mathrm{i}N)^{\ell_{k-1}}\pi_n^{(\ell_{k-1})}(x_{k-1})\Big].$$

If $\ell_i \leq 1, \forall i$, the matrix $W$ is the product of a confluent Vandermonde matrix $V$ and a diagonal scaling $D$, i.e. $W = VD$. If $T(c) \succ 0$ and the numbers of rows of $V$ is greater than its number of columns, the complexity of solving the reformulated dual problem

$$\begin{array}{l} \max \langle b, y\rangle \\ \text{s.t. } (W^*T(c)^{-1}W)^{-1} \succeq \Delta(y), \end{array}$$

depends mostly on the dimension of $\Delta(y)$. That is, an appropriate preprocessing tends to eliminate the dependence on the degree $n$. However, primal strict feasibility cannot be guaranteed from the knowledge of $k$ so that the exact semidefinite relaxation cannot be certified in general.

## 7 Conclusion

Conic optimization problems on several cones of non-negative polynomials, with linear constraints generated by interpolation-like constraints, are studied in this chapter. They naturally induce semidefinite programming problems

$$\begin{array}{l} \min \langle C, X\rangle \\ \text{s.t. } \langle A_i, X\rangle = b_i \; i = 1, \ldots, m \\ \qquad X \succeq 0 \end{array} \tag{54}$$

with low-rank matrices $\{A_i\}_{i=1}^m$. Conditions which guarantee that strict feasibility holds are investigated, see Assumption 3 and the associated discussion. Using Proposition 3, the associated dual problems can be reformulated efficiently; the complexity of solving the reformulated duals is almost independent of the primal space dimension. Finally, new classes of quadratically constrained quadratic programs with exact semidefinite relaxation are described.

## Appendix

The proof presented in Appendix is based on ideas of A. Nemirovskii.

**Proposition 6.** *Let $A = A^*$ be a Hermitian matrix of order $2n+1$. Then the quadratic optimization problem*

$$\begin{array}{l} \min \langle Az, z\rangle \\ \text{s.t. } |z_i| = 1, \; i = 0, \ldots, 2n \end{array} \tag{55}$$

*is* NP*-hard.*

*Proof.* Let $\{a_i\}_{i=0}^n \subseteq \mathbb{Z}$ be a finite set of integers. Checking whether there exist $\{x_i\}_{i=0}^n \subseteq \{-1, +1\}$ such that the equality

$$\sum_{i=0}^{2n} a_i x_i = 0 \tag{56}$$

holds is related to the subset sum problem [4, SP13] and is thus NP-complete.

Let $\{z_\ell\}_{\ell=0}^{2n} \subseteq \mathbb{C}$ be a finite set of complex numbers of modulus one and define the quadratic functions

$$P_\ell(z) = |z_0 - z_{2\ell-1}|^2 + |z_{2\ell-1} - z_{2\ell}|^2 + |z_0 - z_{2\ell}|^2, \quad \ell = 1, \ldots, n.$$

Assume that $z_0$ is equal to 1 without loss of generality. Then the optimization problem

$$\max\{\sum_{\ell=1}^n P_\ell(z) : |z_i| = 1, \forall i\}$$

can be solved explicitly, see Figure 4. Note that the inequality

$$\max\{\sum_{\ell=1}^n P_\ell(z) - |\sum_{\ell=0}^n a_\ell (z_{2\ell+1} - z_{2\ell+2})|^2 : |z_i| = 1, \forall i\}$$
$$\leq \max\{\sum_{\ell=1}^n P_\ell(z) : |z_i| = 1, \forall i\}$$

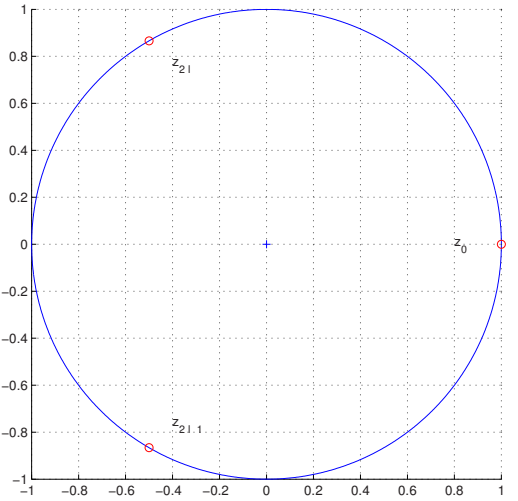



**Fig. 4.** Solution of $\max\{P_\ell(z) : |z_i| = 1, \forall i\}$

is tight if and only if Problem (56) is solvable. Since its left hand side is an instance of (55), this quadratic problem is hard to solve. □

*Acknowledgement.* A research fellowship from the Belgian National Fund for Scientific Research is gratefully acknowledged by the first author. This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

## References

1. B. Alkire and L. Vandenberghe (2002). Convex optimization problems involving finite autocorrelation sequences. Math. Programming, 93:331–359.
2. T. N. Davidson, Z.-Q. Luo, and J. F. Sturm (2002). Linear matrix inequality formulation of spectral mask constraints with applications to FIR filter design. IEEE Trans. Signal Process., 50:2702–2715.
3. S. Feldmann and G. Heinig (1996). Vandermonde factorization and canonical representations of block Hankel matrices. Linear Algebra Appl., 241–243:247–278.
4. M. R. Garey and D. S. Johnson (1979). Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman, San Francisco, CA.
5. Y. Genin, Y. Hachez, Y. Nesterov, and P. Van Dooren (2000). Convex optimization over positive polynomials and filter design. Proc. UKACC International Conference on Control, CD-ROM Paper SS-41, University of Cambridge, UK.
6. Y. Genin, Y. Hachez, Y. Nesterov, and P. Van Dooren (2003). Optimization problems over positive pseudo-polynomial matrices. SIAM J. Matrix Anal. Appl., 25:57–79.

7. Y. Hachez (2003). Convex optimization over non-negative polynomials: structured algorithms and applications. PhD thesis, Université Catholique de Louvain, Department of Mathematical Engineering.
8. Y. Hachez and Y. Nesterov (2002). Optimization problems over nonnegative polynomials with interpolation constraints. Proc. IFAC World Congress, CD-ROM Paper 1650, Barcelona, Spain.
9. T. Kailath and A. H. Sayed (1999, Eds.). Fast reliable algorithms for matrices with structure. SIAM, Philadelphia, PA.
10. S. Karlin and W. J. Studden (1966). Tchebycheff systems: With applications in analysis and statistics. vol. 15 of Pure and Applied Mathematics, Interscience Publishers John Wiley & Sons, New York-London-Sydney.
11. Y. Nesterov (1997). Long-step strategies in interior-point primal-dual methods. Math. Programming, 76:47–94.
12. Y. Nesterov (2000). Squared functional systems and optimization problems. In H.Frenk et al (Editors). High performance optimization. Ch. 17, 405–440, Vol. 33 of Appl. Optim., Kluwer Acad. Publ., Dordrecht.
13. Y. Nesterov and A. Nemirovskii (1994). Interior-point polynomial algorithms in convex programming. Vol. 13 of Studies in Applied Mathematics, SIAM, Philadelphia, PA.
14. A. V. Oppenheim and R. W. Schafer (1989). Discrete-time signal processing. Prentice Hall Signal Processing Series, Prentice Hall, Englewood Cliffs, NJ.
15. G. Pólya and G. Szegő (1976). Problems and theorems in analysis. Vol. II: Theory of functions, zeros, polynomials, determinants, number theory, geometry. Vol. 216 of Die Grundlehren der Mathematischen Wissenschaften, Springer-Verlag, New York.
16. L. Vandenberghe and S. Boyd (1996). Semidefinite programming, SIAM Review, 38:49–95.

# SOSTOOLS and Its Control Applications

Stephen Prajna[1], Antonis Papachristodoulou[1], Peter Seiler[2], and Pablo A. Parrilo[3]

[1] Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA. E-mail: {prajna,antonis}@cds.caltech.edu
[2] Mechanical and Industrial Engineering Dept., University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mail: pseiler@uiuc.edu
[3] Automatic Control Laboratory, Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland. E-mail: parrilo@control.ee.ethz.ch

In this chapter we present SOSTOOLS, a third-party MATLAB toolbox for formulating and solving sum of squares optimization problems. Sum of squares optimization forms a basis for formulating convex relaxations to computationally hard problems such as some that appear in systems and control. Currently, sum of squares programs are solved by casting them as semidefinite programs, which can in turn be solved using interior-point based numerical methods. SOSTOOLS helps this translation in such a way that the underlying computations are abstracted from the user. Here we give a brief description of the toolbox, its features and capabilities (with emphasis on the recently added ones), as well as show how it can be applied to solving problems of interest in systems and control.

## 1 Introduction

There has been a great interest recently in sum of squares polynomials and sum of squares optimization [34, 1, 31, 17, 18, 13, 11], partly due to the fact that these techniques provide convex polynomial time relaxations for many hard problems such as global, constrained, and Boolean optimization, as well as various problem in systems and control. The observation that the sum of squares decomposition can be computed efficiently using semidefinite programming [17] has initiated the development of software tools that facilitate the formulation of the semidefinite programs from their sum of squares equivalents. One such a software is SOSTOOLS [24, 25, 26], a free third-party MATLAB[4] toolbox for solving sum of squares programs.

A multivariate polynomial $p(x_1, \ldots, x_n) \triangleq p(x)$ is a sum of squares (SOS), if there exist polynomials $f_1(x), \ldots, f_m(x)$ such that

[4] A registered trademark of The MathWorks, Inc.

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 273–292, 2005.

$$p(x) = \sum_{i=1}^{m} f_i^2(x). \tag{1}$$

It follows from the definition that the set of sums of squares polynomials in $n$ variables is a convex cone. The existence of an SOS decomposition (1) can be shown to be equivalent to the existence of a positive semidefinite matrix $Q$ such that

$$p(x) = Z^T(x)QZ(x), \tag{2}$$

where $Z(x)$ is the vector of monomials of degree less than or equal to $\deg(p)/2$, i.e., its entries are of the form $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$, where the $\alpha$'s are nonnegative integers and $\alpha_1 + \ldots + \alpha_n \leq \deg(p)/2$. Expressing an SOS polynomial as a quadratic form in (2) has also been referred to as the Gram matrix method [1, 20]. The decomposition (1) can be easily converted into (2) and vice versa. This equivalence makes an SOS decomposition computable using semidefinite programming, since finding a symmetric matrix $Q \succeq 0$ subject to the affine constraint (2) is nothing but a semidefinite programming problem.

It is clear that a sum of squares polynomial is *globally nonnegative*. This is a property of SOS polynomials that is crucial in many control applications, where we replace various polynomial inequalities with SOS conditions. However, it should be noted that not all nonnegative polynomials are necessarily sums of squares. The equivalence between nonnegativity and sum of squares is only guaranteed in three cases, those of univariate polynomials of any even degree, quadratic polynomials in any number of indeterminates, and quartic polynomials in three variables [31]. Indeed nonnegativity is NP-hard to test [12], whereas the SOS conditions are polynomial time verifiable through solving appropriate semidefinite programs. Despite this, in many cases we are able to obtain solutions to computational problems that are otherwise at the moment unsolvable, simply by replacing the nonnegativity conditions with SOS conditions.

A sum of squares program is a convex optimization problem of the following form:

$$\text{Minimize} \sum_{j=1}^{J} w_j c_j \tag{3}$$

subject to

$$a_{i,0}(x) + \sum_{j=1}^{J} a_{i,j}(x) c_j \text{ is SOS, for } i = 1, ..., I, \tag{4}$$

where the $c_j$'s are the scalar real decision variables, the $w_j$'s are given real numbers, and the $a_{i,j}(x)$ are given polynomials (with fixed coefficients). See also another equivalent canonical form of SOS programs in [24, 25]. While the conversion from SOS programs to semidefinite programs (SDPs) can be manually performed for small size instances or tailored for specific problem

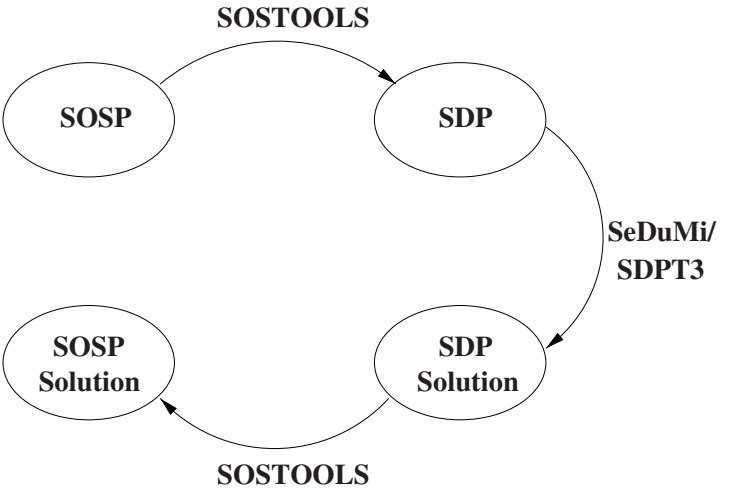



**Fig. 1.** Diagram depicting relations between sum of squares program (SOSP), semidefinite program (SDP), SOSTOOLS, and SeDuMi or SDPT3.

classes, such a conversion can be quite cumbersome to perform in general. It is therefore desirable to have a computational aid that automatically performs this conversion for general SOS programs. This is exactly what SOSTOOLS is useful for. It automates the conversion from SOS program to SDP, calls the SDP solver, and converts the SDP solution back to the solution of the original SOS program (see Figure 1). The current version of SOSTOOLS uses either SeDuMi [35] or SDPT3 [37], both of which are free MATLAB add-ons, as the SDP solver. The user interface of SOSTOOLS has been designed to be as simple, as easy to use, and as transparent as possible, while keeping a large degree of flexibility.

In addition to the optimization problems mentioned above (a related recent software in this regard is GloptiPoly [8], which solves global optimization problems over polynomials, based on the method in [11]), sum of squares polynomials and SOSTOOLS find applications in several control theory problems. These problems include stability analysis of nonlinear, hybrid, and time-delay systems [17, 15, 23, 14], robustness analysis [17, 15, 23], estimation of domain of attraction [17, 33], LPV analysis and synthesis [39], nonlinear synthesis [9, 28, 27], safety verification [22], and model validation [21]. Other areas in which SOSTOOLS is applicable are, for instance, geometric theorem proving [19] and quantum information theory [2].

In Section 2, we present the main features of SOSTOOLS and point out improvements that have been made in the user interface, custom-made functions, and modularity with respect to the choice of semidefinite programming solver. Some control oriented application examples will then be provided in Section 3. In particular, we will consider nonlinear stability analysis, parametric robustness analysis, analysis of time-delay systems, safety verification, and

nonlinear controller synthesis. We show how sum of squares programs corresponding to these problems can be formulated, which in turn can be solved using SOSTOOLS.

## 2 SOSTOOLS Features

It has been mentioned in the introduction that the main purpose of SOSTOOLS is to efficiently transform SOS programs of the form (3)–(4) into semidefinite programs (SDPs), which can then be solved using an SDP solver. The solution is then retrieved from the solver and translated into the original polynomial variables. In this way, the details of the reformulation are abstracted from the user, who can work at the polynomial object level.

Polynomial variables in SOSTOOLS can be defined in two different ways: as a symbolic object through the use of the MATLAB Symbolic Math Toolbox, or as a custom-built polynomial object using the integrated Multivariate Polynomial Toolbox. The former option provides to the user the benefit of making use of all the features in the Symbolic Math Toolbox, which range from simple arithmetic operations to differentiation, integration and polynomial manipulation. On the other hand, the Multivariate Polynomial Toolbox allows users that do not have access to the Symbolic Math Toolbox to use SOSTOOLS. Some basic polynomial manipulation functions are also provided in this toolbox.

To define and solve an SOS program using SOSTOOLS, the user simply needs to follow these steps:

1. Initialize the SOS program.
2. Declare the SOS program variables.
3. Define the SOS program constraints, namely Eq. (4).
4. Set the objective function, namely Eq. (3).
5. Call solver.
6. Get solutions.

We will give a short illustration of these steps in Section 2.1. However, we will not entail in a discussion of how each of these steps is performed nor the SOSTOOLS commands relevant to this. A detailed description can be found in the SOSTOOLS user's guide [25].

In many cases, the SOS program we wish to solve have certain structural properties, such as sparsity, symmetry, and so on. The formulation of the SDP in this case should take into account these properties. This will not only reduce significantly the computational burden of solving it, as the size of the SDP will reduce considerably, but also it removes numerical ill-conditioning. With regard to this, provision has been taken in SOSTOOLS for exploitation of polynomial sparsity when formulating the SDP. The details will be described in Section 2.2.

The frequent use of certain sum of squares programs, such as those corresponding to

- finding the sum of squares decomposition of a polynomial,
- finding lower bounds on polynomial minima, and
- constructing Lyapunov functions for systems with polynomial vector fields

are reflected in the inclusion of customized functions in SOSTOOLS. Some of these customized functions will be discussed at the end of the section.

### 2.1 Formulating Sum of Squares Programs

In the original release of SOSTOOLS, polynomials were implemented solely as symbolic objects, making full use of the capabilities of the MATLAB Symbolic Math Toolbox. This gives to the user the benefit of being able to do all polynomial manipulations using the usual arithmetic operators: `+`, `-`, `*`, `/`, `^`; as well as operations such as differentiation, integration, point evaluation, etc. In addition, it provides the possibility of interfacing with the Maple[5] symbolic engine and the Maple library (which is very advantageous). On the other hand, this prohibited those without access to the Symbolic Toolbox (such as those using the student edition of MATLAB) from using SOSTOOLS. In the current SOSTOOLS release, the user has the option of using an alternative custom-built polynomial object, along with some basic polynomial manipulation methods to represent and manipulate polynomials.

Using the Symbolic Toolbox, a polynomial is created by declaring its independent variables as symbolic variables in the symbolic toolbox and constructing it in a similar way. For example, to create the polynomial $p(x, y) = 2x^2 + 3xy + 4y^4$, one declares the variables $x$ and $y$ by typing

```
>> syms x y
```

and constructs $p(x, y)$ as follows:

```
>> p = 2*x^2 + 3*x*y + 4*y^4
```

In a similar manner, one can define this polynomial using the Multivariate Polynomial Toolbox, a freely available toolbox that has been integrated in SOSTOOLS for constructing and manipulating multivariate polynomials. Polynomial variables are created with the `pvar` command. Here the same polynomial can be constructed by declaring first the variables:

```
>> pvar x y
```

Note that `pvar` is used to replace the command `syms`. New polynomial objects can now be created from these variables, and manipulated using standard addition, multiplication, and integer exponentiation functions:

```
>> p = 2*x^2 + 3*x*y + 4*y^4
```

[5] A registered trademark of Waterloo Maple Inc.

Matrices of polynomials can also be created from polynomials using horizontal/vertical concatenation and block diagonal augmentation. A few additional operations exist in this initial version of the toolbox such as trace, transpose, determinant, differentiation, logical equal, and logical not equal.

The input to the SOSTOOLS commands can be specified using either the symbolic objects or the new polynomial objects. There are some minor variations in performance depending on the degree/number of variables of the polynomials, due the fact that the new implementation always keeps an expanded internal representation, but for most reasonable-sized problems the difference is minimal.

For an illustration, let us now consider the problem of finding a lower bound for the global minimum of the Goldstein-Price test function [5]

$$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]... \\ ... \ [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)].$$

The SOS program for this problem is

$$\begin{aligned} &\text{Minimize } -\gamma, \\ &\text{such that} \\ &\qquad (f(x) - \gamma) \text{ is SOS.} \end{aligned}$$

It is clear that any value of $\gamma$ for which $f(x) - \gamma$ is an SOS will serve as a lower bound for the polynomial, since in that case $f(x) - \gamma$ is nonnegative. By maximizing $\gamma$ (or equivalently, minimizing $-\gamma$), a tighter lower bound can be computed.

In this example, the SOS program is initialized and the decision variable $\gamma$ is declared using the following commands (assuming that we use the polynomial objects)

```
>> pvar x1 x2 gam;
>> prog = sosprogram([x1; x2],gam);
```

The function $f(x)$ is constructed as follows

```
>> f1 = x1+x2+1;
>> f2 = 19-14*x1+3*x1^2-14*x2+6*x1*x2+3*x2^2;
>> f3 = 2*x1-3*x2;
>> f4 = 18-32*x1+12*x1^2+48*x2-36*x1*x2+27*x2^2;
>> f = (1+f1^2*f2)*(30+f3^2*f4);
```

Then the SOS program constraint "$f(x)-\gamma$ is SOS" and the objective function are set, and the SOS program is solved using the following commands

```
>> prog = sosineq(prog,f-gam);
>> prog = sossetobj(prog,-gam);
>> prog = sossolve(prog);
```

The optimal lower bound is then retrieved by

```
>> gam = sosgetsol(prog,gam);
```

The result given by SOSTOOLS is $\gamma = 3$. This is in fact the global minimum of $f(x)$, achieved at $x_1 = 0$, $x_2 = -1$. The same example can also be solved using the customized function `findbound` as follows

```
>> [gam,vars,xopt] = findbound(f);
```

### 2.2 Exploiting Sparsity

The complexity of computing a sum of squares decomposition for a polynomial $p(x)$ depends on two factors: the dimension of the independent variable $x$ and the degree of the polynomial. As mentioned previously, when $p(x)$ has special structural properties, the computation effort can be notably simplified through the reduction of the size of the semidefinite program, removal of degeneracies, and better numerical conditioning.

The first type of simplification can be performed when $p(x)$ is sparse. The notion of sparseness for multivariate polynomials is stronger than the one commonly used for matrices. While in the matrix case this word usually means that many coefficients are zero, in the polynomial case the specific vanishing pattern is also taken into account. This idea is formalized by the concept of *Newton polytope* [36], defined as the convex hull of the set of exponents, considered as vectors in $\mathbb{R}^n$. It was shown by Reznick in [30] that $Z(x)$ need only contain monomials whose squared degrees are contained in the convex hull of the degrees of monomials in $p(x)$. Consequently, for sparse $p(x)$ the size of the vector $Z(x)$ and matrix $Q$ appearing in the sum of squares decomposition can be reduced which results in a decrease of the size of the semidefinite program.

Since the initial version of SOSTOOLS, Newton polytopes techniques have been available via the optional argument `'sparse'` to the function `sosineq`, and in the new release, the support for sparse polynomials has been improved. SOSTOOLS takes the sparse structure into account, and chooses an appropriate set of monomials for the sum of squares decompositions with the convex hull computation performed either by the native MATLAB command `convhulln` (which is based on the software QHULL), or the specialized external package CDD [3]. Special care is taken with the case when the set of exponents has lower affine dimension than the number of variables (this case occurs for instance for homogeneous polynomials, where the sum of the degrees is equal to a constant), in which case a projection to a lower dimensional space is performed prior to the convex hull computation.

A special sparsity structure that appears frequently in robust control theory when considering, for instance, Lyapunov function analysis for linear systems with parametric uncertainty (see Section 3.2), is called the multipartite structure (see [26] for a definition). Such a structure also appears when considering infinitely constrained linear matrix inequalities (LMIs) of the form:

$$\text{Minimize } \sum_{j=1}^{J} w_j c_j$$

subject to

$$A_0(x) + \sum_{j=1}^{J} A_j(x) c_j \succeq 0 \quad \forall x \in \mathbb{R}^n,$$

where the $c_j$'s and $w_j$'s are again the decision variables and given real numbers, respectively, and the $A_j(x)$'s are given symmetric polynomial matrices. By introducing a new set of indeterminates $y$, defining $p_j(x, y) = y^T A_j(x) y$, and replacing the positive semidefiniteness condition $A_0(x) + \sum A_j(x) c_j \succeq 0$ by sum of squares condition

$$p(x, y) = p_0(x, y) + \sum_{j=1}^{J} p_j(x, y) c_j \quad \text{is SOS}, \tag{5}$$

obviously the original LMIs can be computationally relaxed to an SOS program (a positive semidefinite matrix $A(x)$ for which $y^T A(x) y$ is an SOS is called an SOS matrix in [4]). The resulting polynomial (5) has the multipartite structure (in this case it is actually bipartite): its independent variable can be naturally partitioned into two sets $x$ and $y$, where the degree of the polynomial in $x$ can be arbitrary, and the degree in $y$ is always equal to two. What distinguishes this case from a general sparsity, is that the Newton polytope of $p(x, y)$ is the *Cartesian product* of the individual Newton polytopes corresponding to the blocks of variables. Because of this structure, only monomials of the form $x^\alpha y_i$ will appear in the monomial vector $Z(x, y)$. The current version of SOSTOOLS provides a support for the multipartite structure via the argument `'sparsemultipartite'` to the function `sosineq`, by computing a reduced set of monomials in an efficient manner.

To illustrate the benefit of using the sparse multipartite option, consider the problem of checking whether a polynomial matrix inequality

$$F(x) = F^T(x) \succeq 0 \quad \forall x \in \mathbb{R}^n$$

holds, where $F \in \mathbb{R}[x]^{m \times m}$. A sufficient test for positive semidefiniteness of $F(x)$ is obtained by showing that the bipartite polynomial $\mathbf{y}^T F(x) \mathbf{y}$ is a sum of squares (equivalently, showing that $F(x)$ is an SOS matrix). We denote the degree of $F$ by $d$. For various values of $(m, n, d)$, the sizes of the resulting semidefinite programs are depicted in Table 1.

### 2.3 Customized Functions

The SOSTOOLS package includes several "ready-made" customized functions that solve specific problems directly, by internally reformulating them as SOS

**Table 1.** Sizes of the semidefinite programs for proving $F(x) \succeq 0$, where $F \in \mathbb{R}[x]^{m\times m}$ has degree $d$ and $x \in \mathbb{R}^n$, with and without the sparse multipartite option.

| $(m, n, d)$ | Without multipartite option | With multipartite option |
|---|---|---|
| $(3, 2, 2)$ | $15 \times 15$, 90 constraints | $9 \times 9$, 36 constraints |
| $(4, 2, 2)$ | $21 \times 21$, 161 constraints | $12 \times 12$, 60 constraints |
| $(3, 3, 2)$ | $21 \times 21$, 161 constraints | $12 \times 12$, 60 constraints |
| $(4, 3, 2)$ | $28 \times 28$, 266 constraints | $16 \times 16$, 100 constraints |
| $(3, 2, 4)$ | $35 \times 35$, 279 constraints | $18 \times 18$, 90 constraints |
| $(4, 2, 4)$ | $53 \times 53$, 573 constraints | $24 \times 24$, 150 constraints |
| $(3, 3, 4)$ | $59 \times 59$, 647 constraints | $30 \times 30$, 210 constraints |
| $(4, 3, 4)$ | $84 \times 84$, 1210 constraints | $40 \times 40$, 350 constraints |

programs. One of these functions is `findbound`, a function for finding a lower bound of a polynomial, whose usage we have seen at the end of Section 2.1. In the new version, these customized functions have been updated and several new capabilities have been added. For instance, the customized function `findbound`, which previously could only handle unconstrained global polynomial optimization problems, can now be used to solve constrained polynomial optimization problems of the form:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g_i(x) \geq 0, \quad i = 1, ..., M \\ &\qquad\qquad h_j(x) = 0, \quad j = 1, ..., N. \end{aligned}$$

A lower bound for $f(x)$ can be computed using Positivstellensatz-based relaxations. Assume that there exists a set of sums of squares $\sigma_j(x)$'s, and a set of polynomials $\lambda_i(x)$'s, such that

$$\begin{aligned} f(x) - \gamma = \sigma_0(x) + \sum_j \lambda_j(x) h_j(x) + \sum_i \sigma_i(x) g_i(x) + \\ + \sum_{i_1, i_2} \sigma_{i_1,i_2}(x) g_{i_1}(x) g_{i_2}(x) + \cdots \end{aligned} \tag{6}$$

then it follows that $\gamma$ is a lower bound for the constrained optimization problem stated above. This specific kind of representation corresponds to Schmüdgen's theorem [32]. By maximizing $\gamma$, we can obtain a lower bound that becomes increasingly tighter as the degree of the expression (6) is increased.

Another new feature can be found in the customized function `findsos`, which is used for computing an SOS decomposition. For certain applications, it is particularly important to ensure that the SOS decomposition found numerically by SDP methods actually corresponds to a true solution, and is not the result of roundoff errors. This is specially true in the case of ill-conditioned problems, since SDP solvers can sometimes produce in this case unreliable results. There are several ways of doing this, for instance using backwards error

analysis, or by computing rational solutions, that we can fully verify symbolically. Towards this end, we have incorporated an experimental option to round to rational numbers a candidate floating point SDP solution, in such a way to produce an exact SOS representation of the input polynomial (which should have integer or rational coefficients). The procedure will succeed if the computed solution is "well-centered," far away from the boundary of the feasible set; the details of the rounding procedure will be explained elsewhere. Currently, this facility is available only through the customized function `findsos`, by giving an additional input argument `'rational'`. On future releases, we may extend this to more general SOS program formulations.

## 3 Control Applications

We will now see how sum of squares programs can be formulated to solve several problems arising in systems and control, such as nonlinear stability analysis, parametric robustness analysis, stability analysis of time-delay systems, safety verification, and nonlinear controller synthesis.

### 3.1 Nonlinear Stability Analysis

The Lyapunov stability theorem (see e.g. [10]) has been a cornerstone of nonlinear system analysis for several decades. In principle, the theorem states that a system $\dot{x} = f(x)$ with equilibrium at the origin is stable if there exists a positive definite function $V(x)$ such that the derivative of $V$ along the system trajectories is non-positive.

We will now show how the search for a Lyapunov function can be formulated as a sum of squares program. Readers are referred to [17, 15, 23] for more detailed discussions and extensions. For our example, consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_1^3 - x_1 x_3^2 \\ -x_2 - x_1^2 x_2 \\ -x_3 - \frac{3x_3}{x_3^2+1} + 3x_1^2 x_3 \end{bmatrix}, \tag{7}$$

which has an equilibrium at the origin. Notice that the linearization of (7) has zero eigenvalue, and therefore cannot be used to analyze local stability of the equilibrium. Now assume that we are interested in a quadratic Lyapunov function $V(x)$ for proving stability of the system. Then $V(x)$ must satisfy

$$\begin{aligned} V - \epsilon(x_1^2 + x_2^2 + x_3^2) &\geq 0, \\ -\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3 &\geq 0. \end{aligned} \tag{8}$$

The first inequality, with $\epsilon$ being any constant greater than zero (in what follows we will choose $\epsilon = 1$), is needed to guarantee positive definiteness of

$V(x)$. We will formulate an SOS program that computes a Lyapunov function for this system, by replacing the above nonnegativity conditions with SOS conditions. However, notice that $\dot{x}_3$ is a rational function, and therefore (8) is not a polynomial expression. But since $x_3^2 + 1 > 0$ for any $x_3$, we can just reformulate (8) as

$$(x_3^2 + 1)\left(-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3\right) \geq 0.$$

Next, we parameterize the candidate quadratic Lyapunov function $V(x)$ by some unknown real coefficients $c_1$, ..., $c_6$:

$$V(x) = c_1x_1^2 + c_2x_1x_2 + c_3x_2^2 + ... + c_6x_3^2,$$

and the following SOS program (with no objective function) can be formulated

Find a polynomial $V(x)$, (equivalently, find $c_1, ..., c_6$)
such that
$V(x) - (x_1^2 + x_2^2 + x_3^2)$ is SOS,
$(x_3^2 + 1)\left(-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3\right)$ is SOS.

In this example, SOSTOOLS returns $V(x) = 5.5489x_1^2 + 4.1068x_2^2 + 1.7945x_3^2$ as a Lyapunov function that proves the stability of the system.

### 3.2 Parametric Robustness Analysis

When the vector field of the system is uncertain, e.g., dependent on some unknown but bounded parameters $p$, robust stability analysis can be performed by finding a parameter dependent Lyapunov function, which serves as a Lyapunov function for the system for all possible parameter values. Details on computation of such Lyapunov functions can be found in [15, 39].

We will illustrate such robustness analysis by considering the system:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -p_1 & 1 & -1 \\ 2 - 2p_2 & 2 & -1 \\ 3 & 1 & -p_1p_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where $p_1$ and $p_2$ are parameters. The region in the parameter space $(p_1, p_2)$ for which stability is retained is shown in Figure 2. Operating conditions for this system are $p_1 \in [\underline{p_1}, \overline{p_1}]$ and $p_2 \in [\underline{p_2}, \overline{p_2}]$, where $\underline{p_i}, \overline{p_i}$ are real numbers and $\underline{p_i} \leq \overline{p_i}$. We capture this parameter set by constructing two inequalities:

$$a_1(p) \triangleq (p_1 - \underline{p_1})(p_1 - \overline{p_1}) \leq 0$$
$$a_2(p) \triangleq (p_2 - \underline{p_2})(p_1 - \overline{p_2}) \leq 0.$$

We assume that the nominal parameter value is $(2.7, 7)$, which is the center of the rectangular regions shown in Figure 2. The robust stability of this system will be verified by constructing a Lyapunov function. We look for a parameter dependent Lyapunov function of the form $V(x;p)$ that is bipartite: quadratic in the state $x$ and any order in $p$. To ensure that the two Lyapunov inequalities are satisfied in the region of interest, we will adjoint the parameter constraint $a_i(p) \leq 0$ multiplied by sum of squares multipliers $q_{i,j}(x;p)$ to the two Lyapunov inequalities, using a technique that can be considered as an extension of the S-procedure [40]. In this way, the search for a parameter dependent Lyapunov function can be formulated as the following SOS program:

$$\begin{aligned}
&\text{Find } V(x;p), \text{ and } q_{i,j}(x,p),\\
&\text{such that}\\
&V(x;p) - \|x\|^2 + \sum_{j=1}^{2} q_{1,j}(x;p)a_i(p) \text{ is SOS},\\
&-\dot{V}(x;p) - \|x\|^2 + \sum_{j=1}^{2} q_{2,j}(x;p)a_i(p) \text{ is SOS},\\
&q_{i,j}(x;p) \text{ is SOS, for } i,j = 1,2.
\end{aligned}$$

In this case, the Lyapunov function candidate $V(x;p)$ and the sum of squares multiplier $q_{i,j}(x,p)$'s are linearly parameterized by some unknown coefficients, which are the decision variables of our SOS program. We choose the $q_{i,j}(x,p)$'s to be bipartite sums of squares, quadratic in $x$ and of appropriate order in $p$.

When the Lyapunov function $V(x;p)$ is of degree zero in $p$, we can prove stability for $p_1 \in [2.19, 3.21]$ and $p_2 \in [6.47, 7.53]$. When $V(x;p)$ is affine in $p$, then we can prove stability for $p_1 \in [1.7, 3.7]$ and $p_2 \in [5.16, 8.84]$. When it is quadratic in $p$, we can prove stability for the maximum rectangular region centered at the nominal parameter value, i.e., $p_1 \in [1.7, 3.7]$ and $p_2 \in [5, 9]$. See Figure 2.

This example also illustrates the benefit of exploiting the bipartite structure. In the case of quadratic parameter dependence, if the bipartite structure of the conditions is not taken into account then the dimension of the vector $Z(x;p)$ corresponding to a non-structured $V(x;p)$ is 279; taking into account the bipartite structure this number is reduced to 90.

### 3.3 Stability Analysis of Time-Delay Systems

The stability analysis of time-delay systems, i.e., systems described by functional differential equations (FDEs), can be done by constructing appropriate Lyapunov certificates, which are in the form of *functionals* instead of the well known Lyapunov functions that are used in the case of systems described by ordinary differential equations (ODEs). This difference is due to the fact

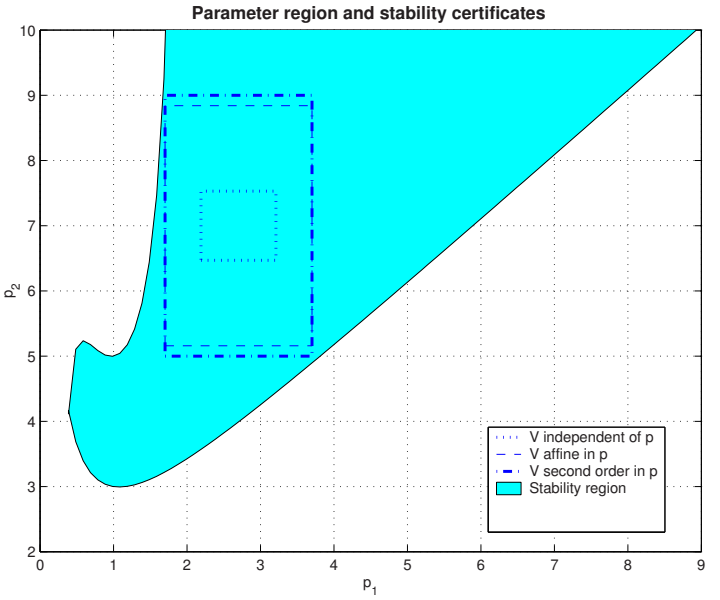



**Fig. 2.** The full stability region (shaded) and the regions for which stability can be proven by constructing bipartite Lyapunov functions using SOSTOOLS. Bigger regions require higher order certificates, which nonetheless can be easily computed because of their structure.

that the state-space in the case of FDEs is the space of functions and not an Euclidean space [7].

Consider a time delay system of the form:

$$\dot{x} = f(x_t), \tag{9}$$

where $x_t = x(t+\theta)$, $\theta \in [-\tau, 0]$ is the *state*. In order to obtain stability conditions for this system, we use the Lyapunov-Krasovskii functional:

$$V(x_t) = a_0(x(t)) + \int_{-\tau}^{0}\int_{-\tau}^{0} a_1(\theta,\xi,x(t),x(t+\theta),x(t+\xi))d\theta d\xi + \\ + \int_{-\tau}^{0}\int_{t+\theta}^{t} a_2(x(\zeta))d\zeta d\theta + \int_{-\tau}^{0}\int_{t+\xi}^{t} a_2(x(\zeta))d\zeta d\xi \tag{10}$$

where by $a_1(\theta,\xi,x(t),x(t+\theta),x(t+\xi))$ we mean a polynomial in $(\theta,\xi,x(t),x(t+\theta),x(t+\xi))$. In the case in which the time delay system is linear, of the form

$$\dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) = f(x_t), \tag{11}$$

the above functional (10) resembles closely the complete Lyapunov functional presented in [6] and we can further restrict ourselves to specially structured kernels, i.e., the polynomial $a_1$ need only be bipartite - quadratic in

$(x(t), x(t+\theta), x(t+\xi))$ but any order in $(\theta, \xi)$. The polynomials $a_0$ and $a_2$ are also quadratic in their arguments. There is also a symmetric structure that should be taken into account:

$$a_1(\theta, \xi, x(t), x(t+\theta), x(t+\xi)) = a_1(\xi, \theta, x(t), x(t+\xi), x(t+\theta)).$$

Here we present the Lyapunov-Krasovskii conditions for stability for concreteness:

**Theorem 1 ([6]).** *The system described by Eq. (11) is asymptotically stable if there exists a bounded quadratic Lyapunov functional $V(x_t)$ such that for some $\epsilon > 0$, it satisfies*

$$V(x_t) \geq \epsilon \|x(t)\|^2 \tag{12}$$

*and its derivative along the system trajectory satisfies*

$$\dot{V}(x_t) \leq -\epsilon \|x(t)\|^2. \tag{13}$$

The Lyapunov-Krasovskii conditions for stability can be satisfied by imposing sums of squares conditions on the *kernels* of the corresponding conditions. There are also extra constraints that have to be added, in the sense that the kernels need to be non-negative only in the integration interval:

$$g_1(\theta) = \theta(\theta + \tau) \leq 0$$
$$g_2(\xi) = \xi(\xi + \tau) \leq 0.$$

Such constraints can be adjoined to the Lyapunov conditions as in the previous example. This yields the following SOS program:

Find polynomials $a_0(x(t)), a_1(\theta, \xi, x(t), x(t+\theta), x(t+\xi)), a_2(x(\zeta)), \epsilon > 0$
and sums of squares $q_{i,j}(\theta, \xi, x(t), x(t+\theta), x(t+\xi))$ for $i, j = 1, 2$
such that

$a_0(x(t)) - \epsilon \|x\|^2$ is SOS,

$$a_1(\theta, \xi, x(t), x(t+\theta), x(t+\xi)) + \sum_{j=1}^{2} q_{1,j} g_j \text{ is SOS},$$

$a_2(x(\zeta))$ is SOS,

$$-\left\{ \begin{array}{c} \frac{da_0}{dx(t)} f(x_t) + \tau^2 \frac{\partial a_1}{\partial x(t)} f(x_t) - \tau^2 \frac{\partial a_1}{\partial \theta} - \tau^2 \frac{\partial a_1}{\partial \xi} + \\ +\tau a_1(0, \xi, x(t), x(t), x(t+\xi)) - \tau a_1(-\tau, \xi, x(t), x(t-\tau), x(t+\xi)) + \\ +\tau a_1(\theta, 0, x(t), x(t+\theta), x(t)) - \tau a_1(\theta, -\tau, x(t), x(t+\theta), x(t-\tau)) + \\ +2\tau a_2(x(t)) - \tau a_2(x(t+\theta)) - \tau a_2(x(t+\xi)) \end{array} \right\} + \ldots$$

$$- \epsilon \|x\|^2 + \sum_{j=1}^{2} q_{2,j} g_j \text{ is SOS}.$$

The first three conditions guarantee positive definiteness of the functional (10) and the last condition guarantees negative definiteness of its time derivative.

**Table 2.** The maximum delay $\tau_{max}$ for different degree polynomials $a_1$ in $\theta$ and $\xi$ corresponding to the example in Section 3.3.

| Order of polynomial $a$ in $\theta$ and $\xi$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $\tau$ | 4.472 | 4.973 | 5.421 | 5.682 | 5.837 | 5.993 | 6.028 |

In order to keep the symmetric and sparse structure in the corresponding sum of squares conditions we have to make a judicious choice for the multipliers $q_{i,j}$.

As an example, consider the following time delay system:

$$\dot{x}_1(t) = -2x_1(t) - x_1(t-\tau) \triangleq f_1$$
$$\dot{x}_2(t) = -0.9x_2(t) - x_1(t-\tau) - x_2(t-\tau) \triangleq f_2.$$

The system is asymptotically stable for $\tau \in [0, 6.17]$. The best bound on $\tau$ that can be obtained with a simple LMI condition is $\tau \in [0, 4.3588]$ in [16]. More complicated LMI conditions that yield better bounds and which are based on a discretization procedure can be found in [6]. Using the Lyapunov functional (10) we get the bounds given in Table 2, where we see that as the order of $a$ with respect to $\theta$ and $\xi$ is increased, better bounds are obtained that approach the analytical one.

The symmetric structure and sparsity of the kernels should be taken into account in the construction of the functionals, as this not only reduces the size of the corresponding semidefinite programs but also removes numerical errors. This can be done using the `'sparsemultipartite'` feature in SOSTOOLS. The construction of Lyapunov functionals can also be extended to uncertain nonlinear systems where delay-dependent and delay-independent conditions can be obtained in a similar manner [14].

### 3.4 Safety Verification

Complex behaviors that can be exhibited by modern engineering systems make the safety verification of such systems both critical and challenging. It is often not enough to design a system to be stable, but a certain bad region in the state space must be completely avoided. Safety verification or reachability analysis aims to show that starting at some initial conditions, a system cannot evolve to an unsafe region in the state space. Here we will show how safety verification can be performed by solving an SOS program, based on what is termed *barrier certificates*. See [22] for detailed discussions and extensions.

For example, let us consider the system (from [22]),

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -x_1 + \frac{1}{3}x_1^3 - x_2,$$

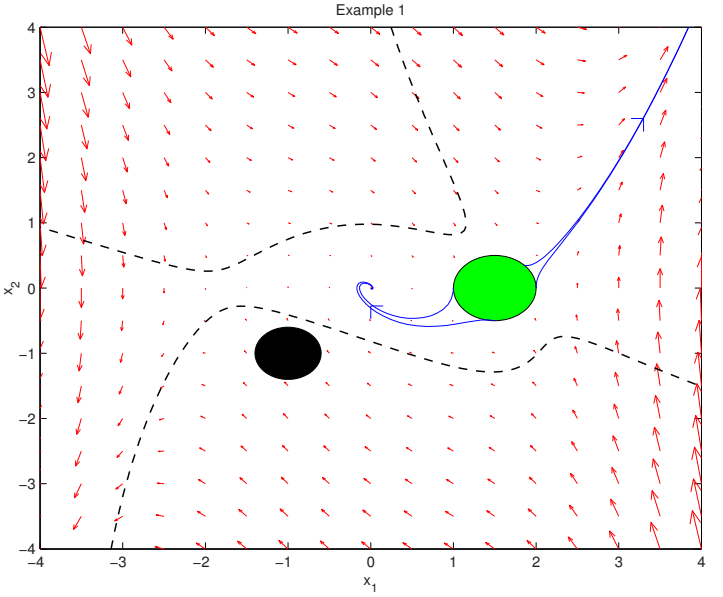



**Fig. 3.** Phase portrait of the system in Section 3.4. Solid patches are (from the left) $\mathcal{X}_u$ and $\mathcal{X}_0$, respectively. Dashed curves are the zero level set of $B(x)$, whereas solid curves are some trajectories of the system.

whose safety we want to verify, with initial set $\mathcal{X}_0 = \{x : g_{\mathcal{X}_0}(x) = (x_1-1.5)^2+x_2^2-0.25 \leq 0\}$ and unsafe set $\mathcal{X}_u = \{x : g_{\mathcal{X}_u}(x) = (x_1+1)^2+(x_2+1)^2-0.16 \leq 0\}$. Here we will find a barrier certificate $B(x)$ which satisfy the following three conditions: $B(x) < 0 \quad \forall x \in X_0$, $B(x) > 0 \quad \forall x \in X_u$, and $\frac{\partial B}{\partial x_1}\dot{x}_1 + \frac{\partial B}{\partial x_2}\dot{x}_2 \leq 0$. It is clear that the existence of such a function guarantees the safety of the system, and the zero level set of $B(x)$ will separate an unsafe region from all system trajectories starting from a given set of initial conditions. By using the higher degree S-procedure and replacing nonnegativity by SOS conditions, we can formulate the following SOS program:

$$
\begin{aligned}
&\text{Find } B(x), \text{ and } \sigma_i(x),\\
&\text{such that } -B(x) - 0.1 + \sigma_1(x) g_{\mathcal{X}_0}(x) \text{ is SOS},\\
&\qquad B(x) - 0.1 + \sigma_2(x) g_{\mathcal{X}_u}(x) \text{ is SOS},\\
&\qquad -\frac{\partial B}{\partial x_1}\dot{x}_1 + \frac{\partial B}{\partial x_2}\dot{x}_2 \text{ is SOS},\\
&\qquad \sigma_i(x) \text{ is SOS, for } i = 1, 2.
\end{aligned}
$$

In this example, we are able to find a quartic barrier certificate $B(x)$ proving the safety of the system, whose zero level set is shown in Figure 3.

### 3.5 Nonlinear Controller Synthesis

For a system $\dot{x} = f(x) + g(x)u$, where $f(x)$ and $g(x)$ are polynomials, application of the SOS technique to the state feedback synthesis problem amounts to finding a polynomial state feedback law $u = k(x)$ and a polynomial Lyapunov function $V(x)$ such that $V(x) - \phi(x)$ and $-\frac{\partial V}{\partial x}(f(x) + g(x)k(x))$ are sums of squares, for some positive definite $\phi(x)$. Yet the set of $V(x)$ and $k(x)$ satisfying these conditions is not jointly convex, and hence a simultaneous search for such $V(x)$ and $k(x)$ is hard — it is equivalent to solving some bilinear matrix inequalities (BMIs). Because of this, a dual approach to the state feedback synthesis based on density functions [29] has also been proposed, which has a better convexity property. The idea in this case is to find a density function $\rho(x)$ and a controller $k(x)$ such that $\rho(x)f(x)/|x|$ is integrable on $\{x \in \mathbb{R}^n : |x| \geq 1\}$ and

$$[\nabla \cdot (\rho(f + gk)](x) > 0 \quad \text{for almost all } x. \tag{14}$$

If such $\rho(x)$ and $k(x)$ can be found, then for almost all initial states $x(0)$ the trajectory $x(t)$ of the closed-loop system exists for $t \in [0, \infty)$ and tends to zero as $t \to \infty$. See [28] for details. It is interesting to note that even if the system is not asymptotically stabilizable, it is sometimes possible to design a controller which makes the zero equilibrium almost globally attractive.

Consider for example the system (taken from [28]).

$$\begin{aligned} \dot{x}_1 &= -6x_1x_2^2 - x_1^2x_2 + 2x_2^3, \\ \dot{x}_2 &= x_2u, \end{aligned}$$

whose zero equilibrium is not asymptotically stabilizable, since any state with $x_2 = 0$ is necessarily an equilibrium. Using the following parameterization

$$\rho(x) = \frac{a(x)}{(x_1^2 + x_2^2)^\alpha}; \quad \rho(x)k(x) = \frac{c(x)}{(x_1^2 + x_2^2)^\alpha},$$

the positivity of $\rho(x)$ and the divergence condition (14) can be formulated as the following SOS program:

$$\begin{aligned} &\text{Find } a(x) \text{ and } c(x), \\ &\text{such that} \\ &\quad a(x) - 1 \text{ is SOS}, \\ &\quad [b\nabla \cdot (fa + gc) - \alpha \nabla b \cdot (fa + gc)](x) \text{ is SOS}, \end{aligned}$$

where $b(x) = x_1^2 + x_2^2$. For $\alpha = 3$, we find that the SOS conditions are fulfilled for $a(x) = 1$ and $c(x) = 2.229x_1^2 - 4.8553x_2^2$. Since the integrability condition is also satisfied, we conclude that the controller $u(x) = \frac{c(x)}{a(x)} = 2.229x_1^2 - 4.8553x_2^2$ renders the origin almost globally attractive. The phase portrait of the closed loop system is shown in Figure 4.

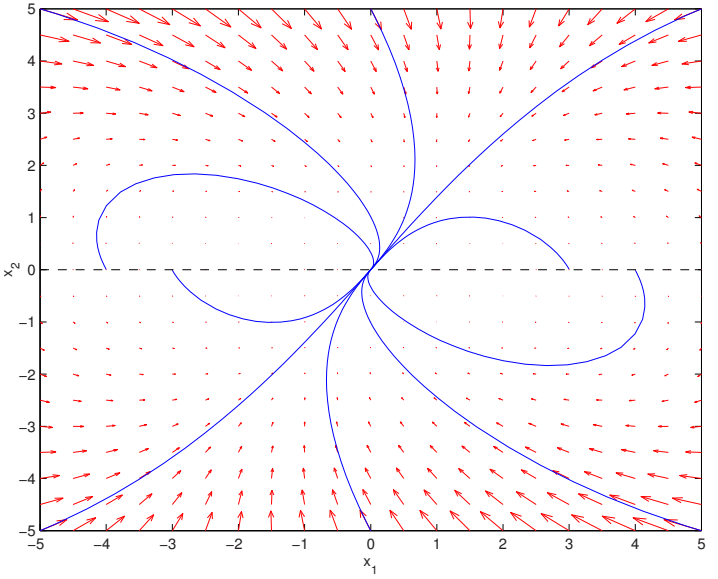



**Fig. 4.** Phase portrait of the closed-loop system in Section 3.5. Solid curves are trajectories; dashed line is the set of equilibria.

## 4 Conclusions

In this chapter we have presented some of the features of SOSTOOLS, a free MATLAB toolbox for formulating and solving SOS programs. We have shown how it can be used to solve some control problems, such as nonlinear stability analysis, parametric robustness analysis, stability analysis of time-delay systems, safety verification, and nonlinear controller synthesis. Future improvements to SOSTOOLS will incorporate symmetry reduction and SOS over quotients, e.g., to handle the case where an SOS decomposition is sought for a polynomial $p(x)$ that is invariant under the action of a finite group.

## References

1. M. D. Choi, T. Y. Lam, and B. Reznick (1995). Sum of squares of real polynomials. Proceedings of Symposia in Pure Mathematics, 58(2):103–126.
2. A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri (2002). Distinguishing separable and entangled states. Physical Review Letters, 88(18).
3. K. Fukuda (2003). CDD/CDD+ reference manual. Available at `www.ifor.math.ethz.ch/staff/fukuda`.
4. K. Gaterman and P. A. Parrilo (2004). Symmetry groups, semidefinite programs, and sums of squares. To appear in Journal of Pure and Appl. Algebra.
5. A. A. Goldstein and J. F. Price (1971). On descent from local minima. Mathematics of Computation, 25:569–574.

6. K. Gu, V. L. Kharitonov, and J. Chen (2003). Stability of Time-Delay systems. Birkhäuser.
7. J. K. Hale and S. M. Verduyn Lunel (1993). Introduction to Functional Differential Equations. Applied Mathematical Sciences (99), Springer-Velag.
8. D. Henrion and J. B. Lasserre (2003). GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi. ACM Transactions on Mathematical Software, 29(2):165–194. Available at `www.laas.fr/~henrion/software/gloptipoly`
9. Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard (2003). Some control applications of sum of squares programming. Proc. IEEE Conference on Decision and Control.
10. H. K. Khalil (1996). Nonlinear Systems. Prentice Hall, Inc., second edition.
11. J. B. Lasserre (2001). Global optimization with polynomials and the problem of moments. SIAM Journal on Optimization, 11(3):796–817.
12. K. G. Murty and S. N. Kabadi (1987). Some NP-complete problems in quadratic and nonlinear programming. Mathematical Programming, 39:117–129.
13. Y. Nesterov (2000). Squared functional systems and optimization problems. In J. Frenk, C. Roos, T. Terlaky, and S. Zhang (Editors). High Performance Optimization, 405–440, Kluwer Academic Publishers.
14. A. Papachristodoulou (2004). Analysis of nonlinear time delay systems using the sum of squares decomposition. Proc. American Control Conference.
15. A. Papachristodoulou and S. Prajna (2002). On the construction of Lyapunov functions using the sum of squares decomposition. Proc. IEEE Conference on Decision and Control.
16. P. G. Park (1999). A delay-dependent stability criterion for systems with uncertain time-invariant delays. IEEE Transactions on Automatic Control, 44(2):876–877.
17. P. A. Parrilo (2000). Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. PhD thesis, California Institute of Technology, Pasadena, CA.
18. P. A. Parrilo (2003). Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming Series B, 96(2):293–320.
19. P. A. Parrilo and R. Peretz (2004). An inequality for circle packings proved by semidefinite programming. Discrete and Computational Geometry, 31(3):357–367.
20. V. Powers and T. Wörmann (1998). An algorithm for sums of squares of real polynomials. Journal of Pure and Applied Linear Algebra, 127:99–104.
21. S. Prajna (2003). Barrier certificates for nonlinear model validation. Proc. IEEE Conference on Decision and Control.
22. S. Prajna and A. Jadbabaie (2004). Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, 477–492, Springer-Verlag.
23. S. Prajna and A. Papachristodoulou (2003). Analysis of switched and hybrid systems – Beyond piecewise quadratic methods. Proc. American Control Conference.
24. S. Prajna, A. Papachristodoulou, and P. A. Parrilo (2002). Introducing SOSTOOLS: A general purpose sum of squares programming solver. Proc. IEEE Conference on Decision and Control.

25. S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo (2004). SOSTOOLS – Sum of Squares Optimization Toolbox, User's Guide, Version 2.00. Available at `www.cds.caltech.edu/sostools` and `control.ee.ethz.ch/~parrilo/sostools`.
26. S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo (2004). New developments in sum of squares optimization and SOSTOOLS. Proc. American Control Conference.
27. S. Prajna, A. Papachristodoulou, and F. Wu (2004). Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach. Proc. Asian Control Conference.
28. S. Prajna, P. A. Parrilo, and A. Rantzer (2004). Nonlinear control synthesis by convex optimization. IEEE Transactions on Automatic Control, 49(2):310–314.
29. A. Rantzer (2001). A dual to Lyapunov's stability theorem. Systems & Control Letters, 42(3):161–168.
30. B. Reznick (1978). Extremal PSD forms with few terms. Duke Mathematical Journal, 45(2):363–374.
31. B. Reznick (2000). Some concrete aspects of Hilbert's 17th problem. In Contemporary Mathematics, 253:251–272, American Mathematical Society.
32. K. Schmüdgen (1991). The $k$-moment problem for compact semialgebraic sets. Mathematische Annalen, 289:203–206.
33. P. Seiler (2003). Stability region estimates for SDRE controlled systems using sum of squares optimization. Proc. American Control Conference.
34. N. Z. Shor (1987). Class of global minimum bounds of polynomial functions. Cybernetics, 23(6):731–734.
35. J. F. Sturm (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 11–12:625–653. Available at `fewcal.kub.nl/sturm/software/sedumi.html`.
36. B. Sturmfels (1998). Polynomial equations and convex polytopes. American Mathematical Monthly, 105(10):907–922.
37. K. C. Toh, R. H. Tütüncü, and M. J. Todd (1999). SDPT3 - a MATLAB software package for semidefinite-quadratic-linear programming. Available at `www.math.nus.edu.sg/~mattohkc/sdpt3.html`.
38. L. Vandenberghe and S. Boyd (1996). Semidefinite programming. SIAM Review, 38(1):49–95.
39. F. Wu and S. Prajna (2004). A new solution approach to polynomial LPV system analysis and synthesis. Proc. American Control Conference.
40. V. A. Yakubovich (1977). S-procedure in nonlinear control theory. Vestnik Leningrad University, 4(1):73–93. English translation.

# Detecting Global Optimality and Extracting Solutions in GloptiPoly

Didier Henrion[1,2] and Jean-Bernard Lasserre[1]

[1] LAAS-CNRS, 7 Avenue du Colonel Roche, 31 077 Toulouse, France.
henrion@laas.fr,lasserre@laas.fr
[2] Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Prague, Czech Republic.

GloptiPoly is a Matlab/SeDuMi add-on to build and solve convex linear matrix inequality (LMI) relaxations of non-convex optimization problems with multivariate polynomial objective function and constraints, based on the theory of moments. In contrast with the dual sum-of-squares decompositions of positive polynomials, the theory of moments allows to detect global optimality of an LMI relaxation and extract globally optimal solutions. In this report, we describe and illustrate the numerical linear algebra algorithm implemented in GloptiPoly for detecting global optimality and extracting solutions. We also mention some related heuristics that could be useful to reduce the number of variables in the LMI relaxations.

## 1 Introduction

Consider the global optimization problem

$$\begin{array}{rl} p^\star = \min_x & g_0(x) \\ \text{s.t.} & g_i(x) \geq 0, \quad i = 1, 2 \ldots, m \end{array} \tag{1}$$

where the mappings $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 0, 1, \ldots, m$ are real-valued polynomials, that is, $g_i \in \mathbb{R}[x_1, \ldots, x_n]$ for all $i = 1, \ldots, m$. Depending on its parity, let $\deg g_i = 2d_i - 1$ or $2d_i$, and denote $d = \max_i d_i$. Define

$$v_k(x) = \begin{bmatrix} 1 & x_1 & x_2 \ \ldots \ x_n & x_1^2 & x_1x_2 \ \ldots \ x_1x_n & x_2^2 & x_2x_3 \ \ldots \ x_n^2 \ \ldots \ x_1^k \ \ldots \ x_n^k \end{bmatrix}^T \tag{2}$$

as a basis for the space of polynomials of degree at most $k$.

A polynomial $g \in \mathbb{R}[x_1, \ldots, x_n]$ can be written

$$x \mapsto g(x) = \sum_{\alpha \in \mathbb{N}^n} g_\alpha x^\alpha$$

D. Henrion and A. Garulli (Eds.): Positive Polynomials in Control, LNCIS 312, pp. 293–310, 2005.

where

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

is a monomial of degree $|\alpha| = \sum_{i=1}^n \alpha_i$.

Following the methodology described in [12], we define for (generally non-convex) problem (1) a hierarchy $\{\mathbb{Q}_k\}$ of (convex) LMI relaxations

$$\mathbb{Q}_k \quad \begin{cases} p_k^* = \min_y \sum_\alpha (g_0)_\alpha y_\alpha \\ \quad \text{s.t. } M_k(y) \succeq 0 \\ \quad\quad M_{k-d_i}(g_i y) \succeq 0, \quad i = 1, 2, \ldots, m \end{cases} \tag{3}$$

where

- each decision variable $y_\alpha$ of $y = \{y_\alpha\}$ corresponds to a monomial $x^\alpha$,
- $M_k(y)$ is the positive semidefinite *moment matrix* of order $k$, and
- $M_{k-d_i}(y)$ is the positive semidefinite *localizing matrix* of order $k - d_i$ associated with the polynomial $g_i$, for all $i = 1, \ldots, m$.

Solving the sequence $\{\mathbb{Q}_k\}$ of LMI relaxations (3) of increasing orders $k = d, d+1, \ldots$, it is proved in [12] that under some mild assumptions on the polynomials $\{g_i\}$, we obtain a monotone sequence of optimal values $p_k^*$ converging asymptotically to the global optimal value $p^*$ of the original optimization problem in (1), i.e. $p_k^* \uparrow p^*$ as $k \to \infty$. Experimental results reveal that in practice $p_k^*$ is very close to $p^*$ for relatively small values of $k$. In addition, in many cases the exact optimal value $p^*$ is obtained at some particular relaxation $\mathbb{Q}_k$, that is, $p^* = p_k^*$ for some relatively small $k$.

GloptiPoly is a user-friendly Matlab/SeDuMi add-on to build and solve these LMI relaxations , see [10] and

`www.laas.fr/~henrion/software/gloptipoly`.

In this report we describe the algorithm used in GloptiPoly to detect whether the global optimum $p^*$ in (1) has been reached at some LMI relaxation $\mathbb{Q}_k$ in (3), i.e. whether $p_k^* = p^*$ for some index $k$. We also describe how to extract (one or several) global minimizers $x^* \in \mathbb{R}^n$ to original problem (1), given a solution $y^*$ of the LMI relaxation $\mathbb{Q}_k$ in (3).

Note that there exist a dual approach to build hierarchy of LMI relaxations, based on real algebraic geometry and sum-of-squares (SOS) decompositions of positive polynomials [16]. In contrast with the theory of moments which works in the space of measures on the primal space of solutions $x \in \mathbb{R}^n$, the SOS approach rather works in a (dual) space of polynomials, to obtain certificates ensuring validity of bounds on the objective function. As a result, and so far, in the latter approach there is no sufficient condition to check whether the exact optimal value is obtained, and no solution extraction mechanism.

In section 2 we state an algebraic condition ensuring global optimality of an LMI relaxation, and we describe the numerical linear algebra algorithm used to extract globally optimal solutions. In section 3 we mention some heuristics based on this algorithm that can used to reduce significantly the number

of variables in the LMI relaxations. Finally, in section 4 we comment on a numerical behavior of GloptiPoly on unconstrained minimization problems. Illustrative numerical examples are inserted throughout the text.

## 2 Extracting Globally Optimal Solutions

### 2.1 Global Optimality Condition

Let $y^*$ be an optimal solution of the LMI relaxation $\mathbb{Q}_k$ in (3) (of order $k$). A sufficient rank condition ensuring global optimality of the LMI relaxation is

$$\text{rank } M_k(y^*) = \text{rank } M_{k-d}(y^*). \tag{4}$$

This condition can be checked numerically with the help of the singular value decomposition [8]. Note however that the rank condition (4) is not necessary, i.e. the global optimum $p^*$ may have been reached at some LMI relaxation of order $k$ (i.e., $p^* = p_k$), and yet rank $M_k(y_k^*) > \text{rank } M_{k-d}(y_k^*)$.

That condition (4) is sufficient to ensure that $p^* = p_k$ is a consequence of a deep result of Curto and Fialkow [6]. In our present context, if condition (4) is true, then by Theorem 1.6 in [6], $y^*$ is the vector of moments of a rank $M_k(y^*)$-atomic measure supported on the feasible set $\mathbb{K} = \{x \in \mathbb{R}^n \,|\, g_i(x) \geq 0,\, i = 1, \ldots, m\}$.

In the important special case where the feasible set $\mathbb{K}$ can be writen

$$\mathbb{K} = \{x \in \mathbb{R}^n \,|\quad g_i(x) = 0,\, i = 1, \ldots, n;\, g_{n+j}(x) \geq 0,\, j = 1, \ldots, m\},$$

and the polynomial ideal $I = \langle g_1, \ldots, g_n \rangle \subset \mathbb{R}[x_1, \ldots, x_n]$ is zero-dimensional and *radical*, then condition (4) is guaranteed to hold at some index $k$. For instance this is the case for boolean (or 0-1) optimization problems, and more generally, bounded discrete optimization problems. For more details the interested reader is referred to [13, 14, 15, 17].

### 2.2 Extraction Algorithm

Assume that the LMI relaxation $\mathbb{Q}_k$ in (3) has been solved, producing a vector $y^*$. Assume further that the rank condition (4) is satisfied. Then the main steps of the extraction algorithm can be sketched as follows.

#### Cholesky Factorization

As condition (4) holds, $y^*$ is the vector of a rank $M_k(y^*)$-atomic measure supported on $\mathbb{K}$. Hence, by construction of the moment matrix $M_k(y^*)$, we have

$$M_k(y^*) = \sum_{j=1}^{r} v_k(x^*(j))(v_k(x^*(j))^T = V^*(V^*)^T$$

where

$$r = \operatorname{rank} M_k(y^*) \tag{5}$$

and

$$V^* = \left[\, v_k(x^*(1))\ v_k(x^*(2)) \cdots\ v_k(x^*(r)) \,\right]$$

where $v_k(x)$ is as in (2), and $\{x^*(j)\}_{j=1}^r$ are $r$ global minimizers of (1).

Extract a Cholesky factor $V$ of the positive semidefinite moment matrix $M_k(y^*)$, i.e. a matrix $V$ with $r$ columns satisfying

$$M_k(y^*) = VV^T. \tag{6}$$

Such a Cholesky factor can be obtained via singular value decomposition, or any cheaper alternative [8].

Matrices $V$ and $V^*$ span the same linear subspace, so the solution extraction algorithm consists in transforming $V$ into $V^*$ by suitable column operations. This is described in the sequel.

**Column Echelon Form**

Reduce matrix $V$ to column echelon form

$$U = \begin{bmatrix} 1 & & & \\ x & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ x & x & x & \\ \vdots & & \ddots & \\ 0 & 0 & 0 & \cdots\ 1 \\ x & x & x & \cdots\ x \\ \vdots & & & \vdots \\ x & x & x & \cdots\ x \end{bmatrix}$$

by Gaussian elimination with column pivoting [8]. By construction of the moment matrix, each row in $U$ corresponds to a monomial $x_\alpha$ in polynomial basis $v$. Pivot elements in $U$ (i.e. the first non-zero elements in each column) correspond to monomials $x_{\beta_j}$, $j = 1, 2, \ldots, r$ of the basis generating the $r$ solutions. In other words, if

$$w = [x_{\beta_1} \quad x_{\beta_2}\ \ldots\ x_{\beta_r}]^T \tag{7}$$

denotes this generating basis, then it holds

$$v = Uw \tag{8}$$

for all solutions $x = x^*(j)$, $j = 1, 2, \ldots, r$.

In summary, extracting the solutions amounts to solving polynomial system of equations (8).

### Solving the Polynomial System of Equations

Once a generating monomial basis is available, it turns out that extracting solutions of polynomial system of equations (8) amounts to solving a linear algebra problem.

As pointed out to us by Monique Laurent, this fact has been rediscovered many times. It is called Stickelberger theorem in textbook [19], and it is credited to Stetter and Müller in [4], see also the recent work [9]. The method can be sketched as follows.

### Multiplication Matrices

For each first degree monomial $x_i$, $i = 1, 2, \ldots, n$ extract from $U$ the $r$-by-$r$ multiplication matrix $N_i$ containing coefficients of monomials $x_i x_{\beta_j}$, $j = 1, 2, \ldots, r$ in generating basis (7), i.e. such that

$$N_i w = x_i w, \quad i = 1, 2, \ldots, n. \tag{9}$$

### Common Eigenvalues

As shown in [19], the entries of solutions $x^*(j)$, $j = 1, 2, \ldots, r$ are common eigenvalues of multiplication matrices $N_i$, $i = 1, 2, \ldots, n$.

In order to compute these eigenvalues, we follow [4] and build a random combination of multiplication matrices

$$N = \sum_{i=1}^{n} \lambda_i N_i$$

where the $\lambda_i$, $i = 1, 2, \ldots, n$ are non-negative real numbers summing up to one. Then, compute the ordered Schur decomposition [8]

$$N = QTQ^T \tag{10}$$

where

$$Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_r \end{bmatrix}$$

is an orthogonal matrix (i.e. $q_i^T q_i = 1$ and $q_i^T q_j = 0$ for $i \neq j$) and $T$ is upper-triangular with eigenvalues of $N$ sorted increasingly along the diagonal.

Finally, the $i$-th entry $x_i^*(j)$ of $x^*(j) \in \mathbb{R}^n$ is given by

$$x_i^*(j) = q_j^T N_i q_j, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, r. \tag{11}$$

### 2.3 Example

Consider the non-convex quadratic optimization problem [12, Ex. 5]

$$\begin{aligned} p^* = \min_x\ & -(x_1-1)^2-(x_1-x_2)^2-(x_2-3)^2 \\ \text{s.t. } & 1-(x_1-1)^2 \geq 0 \\ & 1-(x_1-x_2)^2 \geq 0 \\ & 1-(x_2-3)^2 \geq 0. \end{aligned}$$

Applying the first ($k = 1$) LMI relaxation we obtain $p_1^* = -3$ and rank $M_1(y^*) = 3$.

With the second ($k = 2$) LMI relaxation we obtain $p_2^* = -2$ and rank $M_1(y^*) =$ rank $M_2(y^*) = 3$. Rank condition (4) ensures global optimality, so $p^* = p_2^* = -2$.

The moment matrix of order $k = 2$ reads

$$M_2(y^*) = \begin{bmatrix} 1.0000 & 1.5868 & 2.2477 & 2.7603 & 3.6690 & 5.2387 \\ 1.5868 & 2.7603 & 3.6690 & 5.1073 & 6.5115 & 8.8245 \\ 2.2477 & 3.6690 & 5.2387 & 6.5115 & 8.8245 & 12.7072 \\ 2.7603 & 5.1073 & 6.5115 & 9.8013 & 12.1965 & 15.9960 \\ 3.6690 & 6.5115 & 8.8245 & 12.1965 & 15.9960 & 22.1084 \\ 5.2387 & 8.8245 & 12.7072 & 15.9960 & 22.1084 & 32.1036 \end{bmatrix}$$

and the monomial basis (2) is

$$v_2(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{bmatrix}^T.$$

The Cholesky factor (6) of the moment matrix is given by

$$V = \begin{bmatrix} -0.9384 & -0.0247 & 0.3447 \\ -1.6188 & 0.3036 & 0.2182 \\ -2.2486 & -0.1822 & 0.3864 \\ -2.9796 & 0.9603 & -0.0348 \\ -3.9813 & 0.3417 & -0.1697 \\ -5.6128 & -0.7627 & -0.1365 \end{bmatrix}$$

whose column echelon form reads (after rounding)

$$U = \begin{bmatrix} 1 & \\ 0 & 1 \\ 0 & 0 & 1 \\ -2 & 3 & 0 \\ -4 & 2 & 2 \\ -6 & 0 & 5 \end{bmatrix}.$$

Pivot entries correspond to the following generating basis (7)

$$w = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix}^T.$$

From the subsequent rows in matrix $U$ we deduce from (8) that all the globally optimal solutions $x$ satisfy the polynomial equations

$$\begin{aligned} x_1^2 &= -2 + 3x_1 \\ x_1 x_2 &= -4 + 2x_1 + 2x_2 \\ x_2^2 &= -6 + 5x_2. \end{aligned}$$

Multiplication matrices (9) of monomials $x_1$ and $x_2$ in generating basis $w$ are readily extracted from rows in $U$:

$$N_1 = \begin{bmatrix} 0 & 1 & 0 \\ -2 & 3 & 0 \\ -4 & 2 & 2 \end{bmatrix}, \quad N_2 = \begin{bmatrix} 0 & 0 & 1 \\ -4 & 2 & 2 \\ -6 & 0 & 5 \end{bmatrix}.$$

Then choose e.g.

$$N = 0.6909 N_1 + 0.3091 N_2 = \begin{bmatrix} 0 & 0.6909 & 0.3091 \\ -2.6183 & 2.6909 & 0.6183 \\ -4.6183 & 1.3817 & 2.9274 \end{bmatrix}$$

as a random combination of matrices $N_1$ and $N_2$. The orthogonal matrix in Schur decomposition (10) is given by

$$Q = \begin{bmatrix} 0.4082 & 0.1826 & -0.8944 \\ 0.4082 & -0.9129 & -0.0000 \\ 0.8165 & 0.3651 & 0.4472 \end{bmatrix}.$$

From equations (11), we derive the 3 optimal solutions

$$x^*(1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad x^*(3) = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

### 2.4 Numerical Stability

As shown in [8], all the operations of the solution extraction algorithm are numerically stable, except the Gaussian elimination step with column pivoting. Practical experiments with GloptiPoly however reveal that ill-conditioned problem instances leading to a failure of Gaussian elimination with column pivoting are very scarce. This experimental property of Gaussian elimination was already noticed in [8].

### 2.5 Number of Extracted Solutions

In virtue of relation (5), the number of solutions extracted by the algorithm is equal to the rank of the moment matrix. Up to our knowledge, when solving an LMI relaxation there is no easy way to control the rank of the moment matrix, hence the number of extracted solutions.

If there is no objective function in problem (1), by default GloptiPoly minimizes the trace of the moment matrix. As a result, its rank is indirectly minimized as well. Note however that, in constrast with trace minimization, rank minimization under LMI constraints is a difficult non-convex problem. Practical experiments reveal that low rank moment matrices are preferable from the numerical point of view: they ensure faster convergence to the global optimum. See also example 3.1 for an illustration of the impact of the trace minimization heuristic on the number of extracted solutions.

## 3 Applications of the Extraction Algorithm

When rank condition (4) is not satisfied, then we still can attempt to apply the extraction algorithm described in section 2. If the algorithm is successful and returns feasible solutions reaching the relaxed optimal value $p_k^*$, then by definition of the relaxation $\mathbb{Q}_k$, these solutions are global minimizers. This is the topic of section 3.1. Unfortunately, this heuristic does not work systematically, and extracted solutions can be infeasible, as illustrated with a counterexample in section 3.2.

If the algorithm is not successful, the column echelon form of the Cholesky factor of the moment matrix may contain useful information that can sometimes be exploited to reduce significantly the number of variables, hence the computational burden, in subsequent LMI relaxations. This heuristic is described in section 3.3.

### 3.1 Rank Condition Non Satisfied but Global Optimum Reached

Even though rank condition (4) is not satisfied, the extraction algorithm can be applied successfully, as shown by the following example.

#### Trace Minimization Heuristic

With the help of this example we also return to the comments of section 2.5 on the number of extracted solutions and the trace minimization heuristic.

Consider the polynomial system of equations [4, Ex. 5.2]

$$\begin{aligned} x_1^2 + x_2^2 - 1 &= 0 \\ x_1^3 + (2 + x_3)x_1x_2 + x_2^3 - 1 &= 0 \\ x_3^2 - 2 &= 0. \end{aligned}$$

There is no objective function to be minimized, so as indicated above GloptiPoly solves the LMI relaxations by minimizing the trace of the moment matrix.

Applying the least order ($k = 2$) LMI relaxation we obtain $\text{rank}M_1(y^*) = 4$ and $\text{rank}M_2(y^*) = 7$, so global optimum cannot be ensured via rank condition (4).

With the third LMI relaxation ($k = 3$) we obtain rank $M_1(y^*) =$ rank $M_2(y^*) =$ rank $M_3(y^*) = 2$, so rank condition (4) ensures global optimality.

From the extraction algorithm we derive the two globally optimal solutions

$$x^*(1) = \begin{bmatrix} 0.5826 \\ -0.8128 \\ -1.4142 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} -0.8128 \\ 0.5826 \\ -1.4142 \end{bmatrix}.$$

Now replacing the minimum trace LMI objective function in GloptiPoly with a zero objective function, the third LMI relaxation returns rank$M_1(y^*) =$ 4 and rank $M_2(y^*) =$ rank $M_3(y^*) = 6$, so rank condition (4) cannot ensure global optimality.

However, by applying the extraction algorithm, we are able to extract 6 solutions

$$x^*(1) = \begin{bmatrix} -0.8128 \\ 0.5826 \\ -1.4142 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 0.5826 \\ -0.8128 \\ -1.4142 \end{bmatrix}, \quad x^*(3) = \begin{bmatrix} 0.0000 \\ 1.0000 \\ -1.4142 \end{bmatrix},$$

$$x^*(4) = \begin{bmatrix} 1.0000 \\ 0.0000 \\ -1.4142 \end{bmatrix}, \quad x^*(5) = \begin{bmatrix} 0.0000 \\ 1.0000 \\ 1.4142 \end{bmatrix}, \quad x^*(6) = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 1.4142 \end{bmatrix}$$

thus proving global optimality of the LMI relaxation.

### 3.2 Infeasible Extracted Solutions

When rank condition (4) is not satisfied, it may happen that solutions extracted by the algorithm are infeasible for the original optimization problem. Since solutions are extracted from a convex LMI relaxation, they may be feasible for a subset of the original constraints only.

#### Example

We consider the polynomial systems of equations arising from a test for numerical bifurcation, originally described in [11] and listed in problem collection [2]:

$$\begin{aligned} 5x_1^9 - 6x_1^5 x_2 + x_1 x_2^4 + 2x_1 x_3 &= 0 \\ -2x_1^6 x_2 + 2x_1^2 x_2^3 + 2x_2 x_3 &= 0 \\ x_1^2 + x_2^2 &= 0.265625. \end{aligned}$$

This system has 8 distinct real solutions.

The lowest order ($k = 5$) LMI relaxation yields rank $M_1(y^*) = 3$ and rank $M_2(y^*) =$ rank $M_3(y^*) =$ rank $M_4(y^*) =$ rank $M_5(y^*) = 4$. Since $d = 5$, rank condition (4) cannot ensure global optimality.

The extraction algorithm on moment matrix $M_2(y^*)$ returns 4 solutions

$$x^*(1) = \begin{bmatrix} 0.3653 \\ -0.3636 \\ -0.0153 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 0.3653 \\ 0.3636 \\ -0.0153 \end{bmatrix},$$
$$x^*(3) = \begin{bmatrix} -0.3653 \\ -0.3636 \\ -0.0153 \end{bmatrix}, \quad x^*(4) = \begin{bmatrix} -0.3653 \\ 0.3636 \\ -0.0153 \end{bmatrix}.$$

These solutions satisfy the second and third equations of the original problem, but not the first equation. Indeed, since the solutions are extracted from a convex relaxation of the original problem, they may be infeasible for a subset of the original constraints.

Proceeding with the 6th order LMI relaxation, we obtain $\operatorname{rank} M_i(y^*) = 2$ for all $i = 1, 2, \ldots, 6$, hence ensuring global optimality. The two extracted solutions are

$$x^*(1) = \begin{bmatrix} -0.2619 \\ 0.4439 \\ -0.0132 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 0.2619 \\ 0.4439 \\ -0.0132 \end{bmatrix}.$$

### 3.3 Reducing the Number of LMI Variables

Suppose that at the LMI relaxation of order $k$, equation (8) holds for the solutions to be extracted, i.e. some monomials in standard basis (2) are expressed as linear combinations of monomials of generating basis (7).

If constraints of the original optimization problem become redundant when replacing linearly dependent monomials with combinations of generating monomials, then this results in a reduction of the monomial basis over which subsequent LMI relaxations are built. A similar idea is used in 0-1 quadratic problems to reduce the number of variables in successive LMI relaxations, see [14].

In summary, application of the reduction algorithm at earlier LMI relaxations – at which global optimality cannot be ensured with rank condition (4) – may result in a significant reduction of the problem dimensions. This can be seen as a (heuristic) alternative to the (systematic) algebraic reduction techniques of [7].

### Example with Continuous Variables

Consider the following non-convex quadratic optimization problem suggested by Etienne de Klerk and Radina Dontcheva:

$$\begin{aligned} p^* = \min_x \; & -(x_1-1)^2 - (x_2-1)^2 - (x_3-1)^2 \\ \text{s.t. } & 1-(x_1-1)^2 \geq 0 \\ & 1-(x_2-1)^2 \geq 0 \\ & 1-(x_3-1)^2 \geq 0 \end{aligned} \tag{12}$$

whose global optimum is $p^* = -3$.

At the first ($k = 1$) LMI relaxation, the 4x4 moment matrix $M_1(y^*)$ has rank 4, so obviously no solution can be extracted. However, we obtain $p_1^* = -3$, so the global optimum is reached.

When $k = 2$, we have rank $M_1(y^*) = 4$ and rank $M_2(y^*) = 7$, and the column echelon form of the Cholesky factor of the 10x10 moment matrix $M_2(y^*)$ is given by

$$U = \begin{bmatrix} 1 & & & & & & & & & \\ 0 & 1 & & & & & & & & \\ 0 & 0 & 1 & & & & & & & \\ 0 & 2 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & & & \\ 0 & 0 & 2 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & & & \end{bmatrix}$$

in the monomial basis (2)

$$v_2(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_3 & x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \end{bmatrix}^T.$$

Pivot entries in matrix $U$ correspond to the following generating basis (7)

$$w(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_3 & x_1x_2 & x_1x_3 & x_2x_3 \end{bmatrix}^T$$

which has 7 monomials.

From the rows in matrix $U$ we deduce from (8) that solutions $x$ to be extracted satisfy the polynomial equations

$$\begin{aligned} x_1^2 &= 2x_1 \\ x_2^2 &= 2x_2 \\ x_3^2 &= 2x_3. \end{aligned} \tag{13}$$

The extraction algorithm fails however, because third degree monomials are missing in $U$ to build multiplication matrices (9).

Note however that when substituting monomials as in (13), constraints of the original problem (12) become redundant since $1-(x_i-1)^2 = -x_i^2+2x_i = 0 \geq 0$, for $i = 1, 2, 3$. We can therefore replace monomials $x_i^2$ with $2x_i$ and remove constraints in the next LMI relaxation.

So when $k = 3$, instead of having a basis (2) with 20 monomials of degree 3, we can use only 8 monomials to build the third LMI relaxation – with respect to the previous basis of the second LMI relaxation, the only new element is the third degree monomial $x_1x_2x_3$. Using 8 monomials instead of 20 reduces significantly the computational burden when solving the LMI relaxation. A further reduction is achieved since redundant constraints can be removed and the third LMI relaxation does not feature any localizing matrix.

When applying the reduction algorithm on the moment matrix $M_3(y^*)$ of rank 8, we obtain that monomial $x_1x_2x_3$ belongs to the generating basis. Multiplication matrices are readily obtained, and the 8 expected globally optimal solutions are extracted

$$x^*(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, \quad x^*(3) = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \quad x^*(4) = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix},$$
$$x^*(5) = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}, \quad x^*(6) = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}, \quad x^*(7) = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}, \quad x^*(8) = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}.$$

**Example with Discrete Variables**

Consider the Max-Cut problem

$$\begin{array}{l} \min -\frac{1}{2}\sum_{i<j} w_{ij}(1 - x_ix_j) \\ \text{s.t. } x_i \in \{-1, +1\} \end{array}$$

in the case of a complete K5 graph with adjacency matrix

$$W = \begin{bmatrix} 0\,1\,1\,1\,1 \\ 1\,0\,1\,1\,1 \\ 1\,1\,0\,1\,1 \\ 1\,1\,1\,0\,1 \\ 1\,1\,1\,1\,0 \end{bmatrix}.$$

The first ($k = 1$) LMI relaxation yields $p_1^* = -6.25$ and rank $M_1(y^*) = 5$.

When $k = 2$ we obtain $p_2^* = -6.25$ and rank$M_1(y^*) = 5$, rank$M_2(y^*) = 10$.

When $k = 3$, we get $p_3^* = -6$ and rank $M_1(y^*) = 5$, rank $M_2(y^*) = 10$, rank $M_3(y^*) = 20$. The extraction algorithm returns

$$U = \begin{bmatrix}
1 & & & & & & & & & & & & & & & & & & & \\
0 & 1 & & & & & & & & & & & & & & & & & & \\
0 & 0 & 1 & & & & & & & & & & & & & & & & & \\
0 & 0 & 0 & 1 & & & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 1 & & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & \\
-2 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & -2 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\
0 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{bmatrix}$$

so that linearly dependent monomials in polynomial system of equations (9) are as follows

$$\begin{aligned}
x_4x_5 &= -2 - x_1x_2 - x_1x_3 - x_1x_4 - x_1x_5 - x_2x_3 - x_2x_4 - x_2x_5 - x_3x_4 - x_3x_5 \\
x_1x_2x_3 &= x_4x_5 \\
x_1x_2x_4 &= x_1x_2x_3 \\
x_1x_2x_5 &= x_1x_2x_4 \\
x_1x_3x_4 &= x_1x_2x_5 \\
x_1x_3x_5 &= x_1x_3x_4 \\
x_1x_4x_5 &= -2x_1 - x_2 - x_3 - x_4 - x_5 - x_4x_5 - x_1x_2x_3 - x_1x_2x_4 - x_1x_2x_5 - x_1x_3x_4 \\
x_2x_3x_4 &= -x_1 - x_2 - x_3 - x_4 - x_4x_5 - x_1x_2x_3 - x_1x_2x_5 \\
x_2x_3x_5 &= -x_1 - x_2 - x_3 - x_5 - x_4x_5 - x_1x_2x_4 - x_1x_3x_4 \\
x_2x_4x_5 &= x_1 + x_3 + x_4x_5 + x_1x_2x_5 + x_1x_3x_4 \\
x_3x_4x_5 &= x_1 + x_2 + x_4x_5 + x_1x_2x_3 + x_1x_2x_4.
\end{aligned}$$

From these relations, fourth degree monomials can be expressed in generating basis (7)

$$
\begin{aligned}
x_1x_2x_3x_4 &= (x_1x_2x_3)x_4 = (x_4x_5)x_4 = x_5\\
x_1x_2x_3x_5 &= (x_1x_2x_3)x_5 = (x_4x_5)x_5 = x_4\\
x_1x_2x_4x_5 &= (x_1x_2x_4)x_5 = (x_1x_2x_3)x_5 = (x_4x_5)x_4 = x_5\\
x_1x_3x_4x_5 &= (x_1x_3x_4)x_5 = (x_1x_2x_5)x_5 = x_1x_2\\
x_2x_3x_4x_5 &= (x_2x_3x_4)x_5 = (-x_1 - x_2 - x_3 - x_4 - x_4x_5 - x_1x_2x_3 - x_1x_2x_5)x_5\\
&= -x_1x_5 - x_2x_5 - x_3x_5 - x_4x_5 - 2x_4 - x_1x_2
\end{aligned}
$$

and the only fifth degree monomial readily follows

$$x_1x_2x_3x_4x_5 = (x_1x_2x_3x_4)x_5 = 1.$$

At this stage, it is useless to proceed with higher order LMI relaxations since no more linearly independent monomials of higher degree can be produced.

Consequently, the global optimum $p^* = p_3^* = -6$ has been reached and 20 globally optimal solutions can be extracted from the above matrix $U$.

## 4 A Remark on the Numerical Behavior of GloptiPoly

Finally, we want to comment on a nice and surprising behavior of GloptiPoly that we observed on some examples of unconstrained minimization.

In the case of unconstrained global minimization, that is when $\mathbb{K}$ is $R^n$, only one LMI relaxation is useful, namely $M_k(y)$ if $\deg g_0 = 2k$ or $2k-1$. Indeed,

(a) either $g_0 - p^*$ is SOS and then $p_k^* = p^*$, or
(b) $g_0 - p^*$ is *not* SOS and then $p_{k+j}^* = p_k^* < p^*$ for all $j = 1, 2, \ldots$

Therefore there is no need to try relaxations with orders higher than $k$. However, in case (b) it may be worthy to still try higher order relaxations! Indeed, because of the numerical inaccuracies involved in the solving procedure, one may obtain convergence in a finite number of steps to a value and minimizers, very close to the exact value and the exact minimizers respectively! Let us try to explain why.

If the space of polynomials $x \mapsto g(x) = \sum_\alpha g_\alpha x^\alpha$ is equipped with the norm $\|g\| = \sum_\alpha |g_\alpha|$, then the cone $\Sigma_n$ of SOS polynomials is dense in the set of polynomials nonnegative over the multidimensional box $[-1, 1]^n$, see e.g. [1].

Therefore, consider a nonnegative polynomial $g_0$ that is *not* SOS, and assume that $g_0$ has a global minimizer $x^* \in [-1, 1]^n$ with $g_0(x^*) = p^*$. Then, one may hope that an SOS polynomial $g_k$, *close* to $g_0$ (i.e., with $\|g_k - g_0\| < \epsilon$) will provide a global minimizer close to $x^*$. Observe that for all $x \in [-1, 1]^n$,

$$|g_k(x) - g_0(x)| \;=\; |\sum_\alpha [(g_k)_\alpha - (g_0)_\alpha]\, x^\alpha| \leq \; \|g_k - g_0\| \leq \epsilon.$$

However, one does not know how to construct such a sequence of SOS polynomials $\{g_k\}$ with $\|g_k - g_0\| \to 0$.

But let us see how GloptiPoly behaves on the following well-known example of a non-negative polynomial which is not SOS, namely the polynomial obtained by dehomogenization of Motzkin's form:

$$g_0(x) = \frac{1}{27} + x_1^2 x_2^2 (x_1^2 + x_2^2 - 1).$$

This polynomial is nonnegative ($p^* = \min_x g_0(x) = 0$ attained at $|x_1| = |x_2| = \sqrt{3}/3$) but is not SOS.

- The least order ($k = 3$) LMI relaxation is unbounded, returning no useful information. In principle one should stop here; we have detected that $g_0$ is not SOS.
- when $k = 4$ the LMI relaxation is unbounded too.
- when $k = 5$ the LMI relaxation returns $p_5^* = -0.4036$ and all the moment matrices have full rank (in GloptiPoly we use a relative threshold of $10^{-3}$ to evaluate the numerical rank of a matrix)
- when $k = 6$ the LMI relaxation returns $p_6^* = -0.08241$ and all the moment matrices have full rank
- when $k = 7$ the LMI relaxation returns $p_7^* = -0.01675$ and all the moment matrices have full rank
- when $k = 8$ the LMI relaxation returns an almost zero optimum $p_8^* = 3.022 \cdot 10^{-10}$, and rank $M_1(y^*) = 3$, rank $M_2(y^*) = \text{rank } M_3(y^*) = 4$, thus proving global optimality.

The moment matrix of second order reads

$$M_2(y^*) = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.3333 & 0.0000 & 0.3333 \\ 0.0000 & 0.3333 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.3333 & 0.0000 & 0.0000 & 0.0000 \\ 0.3333 & 0.0000 & 0.0000 & 0.1111 & 0.0000 & 0.1111 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.1111 & 0.0000 \\ 0.3333 & 0.0000 & 0.0000 & 0.1111 & 0.0000 & 0.1111 \end{bmatrix}$$

from which we readily extract the four globally optimal solutions

$$x^*(1) = \begin{bmatrix} -0.5773 \\ -0.5773 \end{bmatrix}, \quad x^*(2) = \begin{bmatrix} 0.5773 \\ -0.5773 \end{bmatrix},$$
$$x^*(3) = \begin{bmatrix} -0.5773 \\ 0.5773 \end{bmatrix}, \quad x^*(4) = \begin{bmatrix} 0.5773 \\ 0.5773 \end{bmatrix}.$$

From the dual LMI [12, 16], we can obtain the SOS decomposition

$$g_8(x) = \sum_{i=1}^{32} a_i^2 q_i^2(x) + \varepsilon r(x) \approx g_0(x) \quad \text{on } [-1, 1]^2,$$

where

- polynomials $q_i(x)$ and $r(x)$ are normalized such that their coefficient vectors have unit Euclidean norm,
- $\varepsilon \leq 10^{-8} < a_i^2$, i.e. positive scalar parameter $\varepsilon$ is less than a given threshold, and positive scalar coefficients $a_i^2$ in the decomposition are greater than the threshold,
- $\deg q_i(x) \leq 8$, since GloptiPoly solved the eighth LMI relaxation,
- there are 32 (!) terms in the SOS decomposition.

The above SOS decomposition is approximate in the sense that parameter $\varepsilon$ is small, but non-zero. It turns out that in GloptiPoly numerical inaccuracy (roundoff errors) helped to find a higher degree SOS polynomial $g_8$ close to Motzkin polynomial on $[-1, 1]^2$.

Thus, everything looks like if in the solving procedure of the dual relaxation $\mathbb{Q}_k^*$ (see [12] for notations) the constraints

$$\langle X, B_\alpha \rangle \;=\; (g_0)_\alpha, \quad |\alpha| \leq 2k,$$

are replaced *automatically* by

$$\langle X, B_\alpha \rangle \;=\; (g_0)_\alpha + \epsilon_\alpha, \quad |\alpha| \leq 2k,$$

with *appropriate* small perturbations $\{\epsilon_\alpha\}$, chosen by the solver!

In a similar vein, it can be useful to add a redundant constraint of the type

$$g_1(x) = R^2 - \|x\|_2^2 \geq 0,$$

and consider the optimization problem $\min\{g_0(x) \,|\, g_1(x) \geq 0\}$, to obtain guaranteed convergence of the successive associated LMI relaxations.

Now consider problem (1) where $g_0(x)$ is the above Motzkin polynomial and $g_1(x)$ is the above radius constraint with $R = 1$ (to include the 4 global minima). With GloptiPoly we obtain already at the third LMI relaxation the SOS decomposition

$$g_0(x) = \sum_{i=1}^{6} a_i^2 q_i^2(x) + g_1(x) \sum_{i=1}^{2} b_i^2 r_i^2(x)$$

with only 6 and 2 terms such that $\deg q_i \leq 3$ and $\deg r_i \leq 2$, respectively.

## 5 Conclusion

Solution extraction is straightforward when the moment matrix has rank-one: in this case the solution vector is equal to the first order moment vector. When the moment matrix has rank greater than one, we have proprosed in section 2 a systematic extraction procedure, implemented in version 2.2 of the GloptiPoly software.

The extraction algorithm is applied when moment matrices satisfy rank condition (4), in which case it is always successful and yields globally optimal solutions. However, as pointed out in section 3, when the rank condition is not satisfied, a heuristic consists in applying the extraction algorithm anyway. Either the algorithm is successful and we are done (see section 3.1) or the algorithm fails, but still some information can be exploited to reduce the number of variables in subsequent LMI relaxations (see section 3.3). Note however that these ideas are not currently implemented in GloptiPoly.

Note finally that an incomplete extraction procedure was sketched in [3] in the case of LMI relaxations for polynomial systems of equations, and partly motivated us to devise a more general algorithm. A specific extraction procedure was also described in [18, Section 5] in the case of quadratic optimization problems with one (possibly non-convex) quadratic constraint, or one linear constraint jointly with one concave quadratic constraint.

**Acknowledgements.**
This work benefited from numerous discussions with Etienne de Klerk, Monique Laurent, Arnold Neumaier and Jos Sturm, whose suggestions, references, and numerical examples are gratefully acknowledged.

## References

1. Berg C (1987). The multidimensional moment problem and semi-groups. Moments in Mathematics. Proc. Symp. Appl. Math., 37:110–124.
2. Bini D, Mourrain B (1998). Polynomial Test Suite. SAGA Project, INRIA Sophia Antipolis, France.
3. Chesi G, Garulli A, Tesi A, Vicino A (2000). An LMI-based Approach for Characterizing the Solution Set of Polynomial Systems. Proc. IEEE Conf. Decision and Control, 1501–1506, Sydney, Australia.
4. Corless R M, Gianni P M, Trager B M (1997). A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. Proc. ACM Int. Symp. Symbolic and Algebraic Computation, 133–140, Maui, Hawaii.
5. Curto R E, Fialkow L A (1991). Recursiveness, positivity, and truncated moment problems. Houston J. Math. 17:603–635.
6. Curto R E, Fialkow L A (2000). The truncated complex $K$-moment problem. Trans. Amer. Math. Soc. 352:2825–2855.
7. Gatermann K, Parrilo P A (2004). Symmetry Groups, Semidefinite Programs, and Sums of Squares. J. Pure and Appl. Algebra, 192(1-3):95–128.
8. Golub G H, Van Loan C F (1996). Matrix computations. 3rd edition. The Johns Hopkins University Press, NY.
9. Jibetean D (2003). Algebraic Optimization with Applications to System Theory. PhD Thesis, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.
10. Henrion D, Lasserre J B (2003). GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi. ACM Trans. Math. Software, 29(2):165–194.

11. Kearfott R B (1987). Some Tests of Generalized Bisection. ACM Trans. Math. Software, 13(3):197–220.
12. Lasserre J B (2001). Global Optimization with Polynomials and the Problem of Moments. SIAM J. Opt. 11(3):796–817.
13. Lasserre J B (2001). Polynomials nonnegative on a grid and discrete optimization. Trans. Amer. Math. Soc. 354:631–649.
14. Lasserre J B (2002). An Explicit Equivalent Positive Semidefinite Program for Nonlinear 0-1 Programs. SIAM J. Opt. 12(3):756–769.
15. Laurent M (2002). Semidefinite representations for finite varieties. Preprint. Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 2002.
16. Parrilo P A (2003). Semidefinite Programming Relaxations for Semialgebraic Problems. Math. Prog. Ser. B. 96:293–320.
17. Parrilo P A (2002). An explicit construction of distinguished representations of polynomials nonnegative over finite sets. Preprint. ETH Zurich, Switzerland.
18. Sturm J F, Zhang S (2003). On Cones of Nonnegative Quadratic Functions. Math. Op. Res. 28:246–267.
19. Sturmfels B (2002). Solving Systems of Polynomial Equations. Amer. Math. Soc., Providence, RI.

# Index


$H_\infty$, 48, 73, 75, 76, 82

analysis
  constrained system, 30
  disturbance, 7, 10, 11, 13, 14
  time-delay systems, 284

Bernstein's algorithm, 172
Bilinear Matrix Inequality, 49

central path, 226
Cholesky factor, 296, 298, 300, 303
column echelon form, 296, 298, 300
complementary slackness, 222
conic relaxations, 124
conjugate gradients, 197
constraint qualification, 51, 53
controllability, 56, 207, 232
convex optimization, 239, 244
copositive matrices, 123
cutting-plane method, 197

dual SDP, 204
duality, 54, 204
  gap, 222

energy density spectrum, 251
exponential nonlinearity, 39

Fejér-Riesz theorem, 243
filter design, 200
fixed-order controller, 48, 73, 74
form, 121, 125
  even, 125, 128
  homogeneous, 88, 90
  positive on the simplex, 123
frequency-domain inequality, 199

gain-scheduling, 103, 203
generating basis, 296–299, 302–305
global optimization, 293
global optimum, 294, 295
GloptiPoly, 293, 294, 299, 306
Goursat transform, 159
Gröbner bases, 186

Hankel matrix, 242, 262
Hessian, 129
Hilbert, 154

ideal, 5, 186
integral quadratic constraints (IQCs), 200
interior-point algorithm, 204
  general-purpose, 204–210
  implementation of, 204
  primal-dual, 204, 222
interpolation constraints, 239, 245, 246
irrational nonlinearity, 38

Kalman-Yakubovich-Popov lemma, 112, 195
KYP-SDP, 195–235

Lagrange polynomial, 242, 244
Linear matrix inequality, *see* LMI
linear programming, 227
linear-quadratic regulator, 201

LMI, 73, 74, 78, 80, 82, 88, 92, 94–96, 99, 195
  infinitely constrained, 280
  relaxations, 61, 294, 295, 298–300, 302–304, 306, 308, 309
localizing matrix, 294, 303
LPV systems, 103
Lyapunov
  equation, 209, 230
  function, 87, 99, 202
  function domain of attraction, 191
  function parameter-dependent, 88, 99, 103, 283
  stability, 27

Markov-Lukács theorem, 241
moment matrix, 294–296, 298–300, 302–304, 307, 308
monoid, 5
monomial, 4
Motzkin, 156
  form, 307
  polynomial, 308
moving average system, 251
multipartite structure, 279
multiplication matrix, 297, 299, 304

Nesterov-Todd scaling, 226
Newton equations, 204, 226, 228
Newton polytope, 185, 279
non-polynomial vector field, 29
nonlinear controller synthesis, 289
nonlinear stability analysis, 282

Pólya's theorem, 159
Pólya's theorem, 123
parametric robustness analysis, 283
parametric uncertainty, 87, 92
PID control, 47, 48
polynomial, 4, 73
  compact level set, 129
  even, 125
  homogeneous, 121
  inequalities, 151
  matrix, 49, 78
  non-negative, 239
  positive, 73, 294
  positive on the unit sphere, 123
  sparsity, 279
  system of equations, 296, 300, 301, 305, 309
  uncertain two–variable, 169
  with nonnegative coefficients, 125
Positivstellensatz, 5, 184, 281
  Putinar's, 123
  Schmüdgen's, 123
Putinar, 50, 151
  positivstellensatz, 123

quadratic module, 151

real algebraic geometry, 181
recasting, 29
relaxations, 61
Riccati equation, 201
robust analysis, 57
robust control, 200, 215
robust parametric margin, 96, 98
robust stability, 87, 90, 94, 96, 99, 173, 200, 203

$\mathcal{S}$-procedure, 6
safety verification, 287
saturation nonlinearity, 34
scalarization, 58
Schmüdgen's positivstellensatz, 123
Schur
  complement, 255
  decomposition, 297, 299
SDPT3, 217, 223
SeDuMi, 217
semialgebraic set, 151, 181
semidefinite program, 49, 203, 222
  dual, 204
  optimality conditions, 222
simplex, 121
singular value decomposition, 295, 296
SOS, *see* sum of squares
SOSTOOLS, 27, 186, 273
sparsity, 185, 279
SPR, 73
stability of 2D polynomials, 165
stabilization, 103
state feedback, 15
  algorithm, 17
  example, 19
  with saturation, 18
sum of squares, 50, 109, 125, 129, 151, 294

decomposition, 26, 274
matrix, 50, 59
on invariant rings, 190
on quotient rings, 187
polynomial, 4
program, 28, 274
programming, 6
via SDP, 183
symmetries, 188
Toeplitz matrix, 244
trigonometric nonlinearity, 36

Vandermonde
confluent matrix, 267
factorization, 242
matrix, 242, 246, 247, 261

Youla-Kučera parametrization, 74, 199

# Lecture Notes in Control and Information Sciences

## Edited by M. Thoma and M. Morari

**Vol. 283:** Fielding, Ch. et al. (Eds)
Advanced Techniques for Clearance of
Flight Control Laws
480 p. 2003 [3-540-44054-2]

**Vol. 282:** Schröder, J.
Modelling, State Observation and
Diagnosis of Quantised Systems
368 p. 2003 [3-540-44075-5]

**Vol. 281:** Zinober A.; Owens D. (Eds)
Nonlinear and Adaptive Control
416 p. 2002 [3-540-43240-X]

**Vol. 280:** Pasik-Duncan, B. (Ed)
Stochastic Theory and Control
564 p. 2002 [3-540-43777-0]

**Vol. 279:** Engell, S.; Frehse, G.; Schnieder, E. (Eds)
Modelling, Analysis, and Design of Hybrid Systems
516 p. 2002 [3-540-43812-2]

**Vol. 278:** Chunling D. and Lihua X. (Eds)
$H_\infty$ Control and Filtering of
Two-dimensional Systems
161 p. 2002 [3-540-43329-5]

**Vol. 277:** Sasane, A.
Hankel Norm Approximation
for Infinite-Dimensional Systems
150 p. 2002 [3-540-43327-9]

**Vol. 276:** Bubnicki, Z.
Uncertain Logics, Variables and Systems
142 p. 2002 [3-540-43235-3]

**Vol. 275:** Ishii, H.; Francis, B.A.
Limited Data Rate in Control Systems with Networks
171 p. 2002 [3-540-43237-X]

**Vol. 274:** Yu, X.; Xu, J.-X. (Eds)
Variable Structure Systems:
Towards the $21^{\mathrm{st}}$ Century
420 p. 2002 [3-540-42965-4]

**Vol. 273:** Colonius, F.; Grüne, L. (Eds)
Dynamics, Bifurcations, and Control
312 p. 2002 [3-540-42560-9]

**Vol. 272:** Yang, T.
Impulsive Control Theory
363 p. 2001 [3-540-42296-X]

**Vol. 271:** Rus, D.; Singh, S.
Experimental Robotics VII
585 p. 2001 [3-540-42104-1]

**Vol. 270:** Nicosia, S. et al.
RAMSETE
294 p. 2001 [3-540-42090-8]

**Vol. 269:** Niculescu, S.-I.
Delay Effects on Stability
400 p. 2001 [1-85233-291-316]

**Vol. 268:** Moheimani, S.O.R. (Ed)
Perspectives in Robust Control
390 p. 2001 [1-85233-452-5]

**Vol. 267:** Bacciotti, A.; Rosier, L.
Liapunov Functions and Stability in Control Theory
224 p. 2001 [1-85233-419-3]

**Vol. 266:** Stramigioli, S.
Modeling and IPC Control of Interactive Mechanical
Systems – A Coordinate-free Approach
296 p. 2001 [1-85233-395-2]

**Vol. 265:** Ichikawa, A.; Katayama, H.
Linear Time Varying Systems and Sampled-data Systems
376 p. 2001 [1-85233-439-8]

**Vol. 264:** Baños, A.; Lamnabhi-Lagarrigue, F.;
Montoya, F.J
Advances in the Control of Nonlinear Systems
344 p. 2001 [1-85233-378-2]

**Vol. 263:** Galkowski, K.
State-space Realization of Linear 2-D Systems with
Extensions to the General nD ($n > 2$) Case
248 p. 2001 [1-85233-410-X]

**Vol. 262:** Dixon, W.; Dawson, D.M.; Zergeroglu, E.;
Behal, A.
Nonlinear Control of Wheeled Mobile Robots
216 p. 2001 [1-85233-414-2]

**Vol. 261:** Talebi, H.A.; Patel, R.V.; Khorasani, K.
Control of Flexible-link Manipulators
Using Neural Networks
168 p. 2001 [1-85233-409-6]

**Vol. 260:** Kugi, A.
Non-linear Control Based on Physical Models
192 p. 2001 [1-85233-329-4]

**Vol. 259:** Isidori, A.; Lamnabhi-Lagarrigue, F.;
Respondek, W. (Eds)
Nonlinear Control in the Year 2000 Volume 2
640 p. 2001 [1-85233-364-2]

**Vol. 258:** Isidori, A.; Lamnabhi-Lagarrigue, F.;
Respondek, W. (Eds)
Nonlinear Control in the Year 2000 Volume 1
616 p. 2001 [1-85233-363-4]

**Vol. 257:** Moallem, M.; Patel, R.V.; Khorasani, K.
Flexible-link Robot Manipulators
176 p. 2001 [1-85233-333-2]