

# Towards Scalable Algorithms with Formal Guarantees for Lyapunov Analysis of Control Systems via Algebraic Optimization

(Tutorial paper for the 53rd IEEE Conference on Decision and Control)

Amir Ali Ahmadi and Pablo A. Parrilo

**Abstract**—Exciting recent developments at the interface of optimization and control have shown that several fundamental problems in dynamics and control, such as stability, collision avoidance, robust performance, and controller synthesis can be addressed by a synergy of classical tools from *Lyapunov theory* and modern computational techniques from *algebraic optimization*. In this paper, we give a brief overview of our recent research efforts (with various coauthors) to (i) enhance the scalability of the algorithms in this field, and (ii) understand their worst case performance guarantees as well as fundamental limitations. Our results are tersely surveyed and challenges/opportunities that lie ahead are stated.

## I. ALGEBRAIC METHODS IN OPTIMIZATION AND CONTROL

In recent years, a fundamental and exciting interplay between *convex optimization* and *algorithmic algebra* has allowed for the solution or approximation of a large class of nonlinear and nonconvex problems in optimization and control once thought to be out of reach. The success of this area stems from two facts: (i) Numerous fundamental problems in optimization and control (among several other disciplines in applied and computational mathematics) are *semialgebraic*; i.e., they involve optimization over sets defined by a finite number of possibly quantified *polynomial inequalities*. (ii) Semialgebraic problems can be reformulated as optimization problems over the set of *nonnegative polynomials*. This makes them amenable to a rich set of algebraic tools which lend themselves to *semidefinite programming*—a subclass of convex optimization problems for which global solution methods are available.

Application areas within optimization and computational mathematics that have been impacted by advances in algebraic techniques are numerous: approximation algorithms for NP-hard combinatorial problems [1], equilibrium analysis of continuous games [2], robust and stochastic optimization [3], statistics and machine learning [4], software verification [5], filter design [6], quantum computation [7], and automated theorem proving [8], are only a few examples on a long list.

In dynamics and control, algebraic methods and in particular the so-called area of “*sum of squares (sos) optimization*” [9], [10], [11], [12], [13] have rejuvenated Lyapunov theory, giving the hope or the outlook of a paradigm shift

from classical linear control to a principled framework for design of nonlinear (polynomial) controllers that are provably safer, more agile, and more robust. As a concrete example, Figure 1 demonstrates our recent work with Majumdar and Tedrake [14] in this area applied to the field of robotics. As the caption explains, sos techniques provide controllers with much larger margins of safety along planned trajectories and can directly reason about the *nonlinear* dynamics of the system under consideration. These are crucial assets for more challenging robotic tasks such as walking, running, and flying. Sum of squares methods have also recently made their way to actual industry flight control problems, e.g., to explain the falling leaf mode phenomenon of the F/A-18 Hornet aircraft [15], [16] or to design controllers for hypersonic aircraft [17].

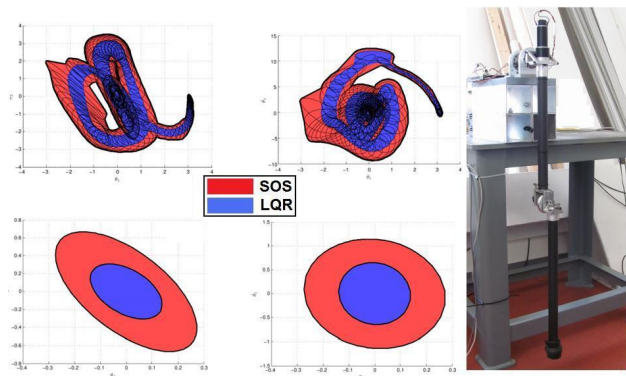


Fig. 1. From [14] (with Majumdar and Tedrake): The “swing-up and balance” task via sum of squares optimization for an underactuated and severely torque limited double pendulum (the “Acrobot”). Top: projections of basins of attraction around a nominal swing-up trajectory designed by linear quadratic regulator (LQR) techniques (blue) and by SOS techniques (red). Bottom: projections of basins of attraction of the unstable equilibrium point in the upright position stabilized by a linear controller via LQR (blue), and a cubic controller via SOS (red). To our knowledge, this work constitutes the first hardware implementation and experimental validation of sum of squares techniques in robotics.

## II. OUR TARGET AREAS IN ALGEBRAIC OPTIMIZATION AND CONTROL

Despite the wonderful advances in algebraic techniques for optimization and their successful interplay with Lyapunov methods, there are still many fundamental challenges to

overcome and unexplored pathways to pursue. In this paper, we aim at highlighting two concrete areas in this direction:

**Area 1—Struggle with scalability:** Scalability is arguably the single most outstanding challenge for algebraic methods, not just in control theory, but in all areas of computational mathematics where these techniques are being applied today. It is well known that the size of the semidefinite programs (SDPs) resulting from sum of squares techniques (although polynomial in the data size) grows quickly and this limits the scale of the problems that can be efficiently and reliably solved with available SDP solvers. This drawback deprives large-scale systems of the application of algebraic techniques and perhaps equally importantly shuts the door on the opportunities that lie ahead if we could use these tools for *real-time optimization*.

In nonlinear control, problems with scalability also manifest themselves in form of complexity of Lyapunov functions. It is common for “simple” (e.g., low degree) stable systems to necessitate “complicated” Lyapunov functions as stability certificates (e.g., polynomials of high degree). The more complex the Lyapunov function, the more variables its parametrization will have, and the larger the sum of squares programs that search for it will be. In view of this, it is of particular interest to derive conditions for stability that are less stringent than those of classical Lyapunov theory. A related challenge in this area is the lack of a unified and comparative theory for various classes of Lyapunov functions available in the literature (e.g., polytopic, piecewise quadratic, polynomial, etc.). These problems are more pronounced in the study of *uncertain* or *hybrid system*, which are of great practical relevance.

**Area 2—Lack of rigorous guarantees:** While most works in the literature formulate hierarchies of optimization problems that—if feasible—guarantee desired properties of a control system of interest (e.g., stability or safety), relatively few establish “converse results”, i.e., proofs that if certain policies meet design specifications, then a particular level in the optimization hierarchy is guaranteed to find a certificate as a feasible solution. This is in contrast to more discrete areas of optimization where tradeoffs between algorithmic efficiency and worst-case performance guarantees are often quite well-understood.

A study of performance guarantees for some particular class of algorithms (in our case, sum of squares algorithms) naturally borders the study of lower bounds, i.e., fundamental limits on the efficiency of *any* algorithm that provably solves a problem class of interest. Once again here, the state of affairs in this area of controls is not entirely satisfactory: there are numerous fundamental problems in the field that while believed to be “hard” in folklore, lack a rigorous complexity-theoretic lower bound. One can attribute this shortcoming to some extent to the nature of most problems in controls, which typically come from continuous mathematics and at times describe qualitative behavior of a system rather than quantitative ones (consider, e.g., asymptotic stability of a nonlinear vector field).

The remainder of this paper, whose goal is to accompany

our tutorial talk, presents a brief report on some recent progress we have made on these two target areas, as well as some challenges that lie ahead. This is meant neither as a comprehensive survey paper, as there are many great contributions by other authors that we do not cover, nor as a stand-alone paper, as for the most part only entry points to a collection of relevant papers will be provided. The interested reader can find further detail and a more comprehensive literature review in the references presented in each section.

### A. Organization of the paper

The outline of the paper is as follows. We start by a short section on basics of sum of squares optimization in the hope that our tutorial paper becomes accessible to a broader audience. In Section IV, we describe some recent developments on the optimization side to provide more scalable alternatives to sum of squares programming. This is the framework of “*dsos and sdsos optimization*”, which is amenable to linear and second order cone programming as opposed to semidefinite programming. In Section V, we describe some new contributions to Lyapunov theory that can improve the scalability of algorithms meant for verification of dynamical systems. These include techniques for replacing high-degree Lyapunov functions with multiple low-degree ones (Section V-A), and a methodology for relaxing the “monotonic decrease” requirement of Lyapunov functions (Section V-B). The beginning of Section V also includes a list of recent results on complexity of deciding stability and on success/limitations of algebraic methods for finding Lyapunov functions. Both Sections IV and V are ended with a list of open problems or opportunities for future research.

## III. A QUICK INTRODUCTION TO SOS FOR THE GENERAL READER<sup>1</sup>

At the core of most algebraic methods in optimization and control is the simple idea of optimizing over polynomials that take only *nonnegative* values, either globally or on certain regions of the Euclidean space. A multivariate polynomial  $p(x) := p(x_1, \dots, x_n)$  is said to be (globally) *nonnegative* if  $p(x) \geq 0$  for all  $x \in \mathbb{R}^n$ . As an example, consider the task of deciding whether the following polynomial in 3 variables and degree 4 is nonnegative:

$$p(x) = x_1^4 - 6x_1^3x_2 + 2x_1^3x_3 + 6x_1^2x_2^2 + 9x_1^2x_2x_3 - 6x_1^2x_2x_3 - 14x_1x_2x_3^2 + 4x_1x_3^3 + 5x_3^4 - 7x_2^2x_3^2 + 16x_2^4. \quad (1)$$

This may seem like a daunting task (and indeed it is as testing for nonnegativity is NP-hard), but suppose we could “somehow” come up with a decomposition of the polynomial as a sum of squares:

$$p(x) = (x_1^2 - 3x_1x_2 + x_1x_3 + 2x_3^2)^2 + (x_1x_3 - x_2x_3)^2 + (4x_2^2 - x_3^2)^2. \quad (2)$$

<sup>1</sup>The familiar reader may safely skip this section. For a more comprehensive introductory exposition, see:

<https://blogs.princeton.edu/imabandit/guest-posts/>

Then, we have at our hands an *explicit algebraic certificate* of nonnegativity of  $p(x)$ , which can be easily checked (simply by multiplying the terms out). A polynomial  $p$  is said to be a *sum of squares* (sos), if it can be written as  $p(x) = \sum q_i^2(x)$  for some polynomials  $q_i$ . Because of several interesting connections between real algebra and convex optimization discovered in recent years [18] and quite well-known by now, the question of existence of an sos decomposition (i.e., the task of going from (1) to (2)) can be cast as a *semidefinite program* (SDP) and be solved, e.g., by interior point methods. The question of *when* nonnegative polynomials admit a decomposition as a sum of squares is one of the central questions of real algebraic geometry, dating back to the seminal work of Hilbert [19], [20], and an active area of research today. This question is commonly faced when one attempts to prove guarantees for performance of algebraic algorithms in optimization and control.

In short, sum of squares decomposition is a sufficient condition for polynomial nonnegativity. It has become quite popular because of three reasons: (i) the decomposition can be obtained by semidefinite programming, (ii) the proof of nonnegativity is in form of an *explicit certificate* and is easily verifiable, and (iii) there is strong empirical (and in some cases theoretical) evidence showing that in relatively low dimensions and degrees, “most” nonnegative polynomials are sums of squares.

But why do we care about polynomial nonnegativity to begin with? We briefly present two fundamental application areas next: the polynomial optimization problem, and Lyapunov analysis of control systems.

### A. The polynomial optimization problem

The polynomial optimization problem (POP) is currently a very active area of research in the optimization community. It is the following problem:

$$\begin{aligned} & \text{minimize} && p(x) \\ & \text{subject to} && x \in K := \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, h_i(x) = 0\}, \end{aligned} \quad (3)$$

where  $p$ ,  $g_i$ , and  $h_i$  are multivariate polynomials. The special case of problem (3) where the polynomials  $p$ ,  $g_i$ ,  $h_i$  all have degree one is of course *linear programming*, which can be solved very efficiently. When the degree is larger than one, POP contains as special case many important problems in operations research; e.g., all problems in the complexity class NP, such as MAXCUT, travelling salesman, computation of Nash equilibria, scheduling problems, etc.

A set defined by a finite number of polynomial inequalities (such as the set  $K$  in (3)) is called *basic semialgebraic*. By a straightforward reformulation of problem (3), we observe that if we could optimize over the set of polynomials, *nonnegative on a basic semialgebraic set*, then we could solve the POP problem to global optimality. To see this, note that the optimal value of problem (3) is equal to the optimal value of the following problem:

$$\begin{aligned} & \text{maximize} && \gamma \\ & \text{subject to} && p(x) - \gamma \geq 0, \quad \forall x \in K. \end{aligned} \quad (4)$$

Here, we are trying to find the largest constant  $\gamma$  such that the polynomial  $p(x) - \gamma$  is nonnegative on the set  $K$ ; i.e., the largest lower bound on problem (3). For ease of exposition, we only explained how a sum of squares decomposition provides a sufficient condition for polynomial nonnegativity globally. But there are straightforward generalizations for giving sos certificates that ensure nonnegativity of a polynomial on a basic semialgebraic set; see, e.g., [18]. All these generalizations are amenable to semidefinite programming and commonly used to tackle the polynomial optimization problem.

### B. Lyapunov analysis of dynamical systems

Numerous fundamental problems in nonlinear dynamics and control, such as stability, invariance, robustness, collision avoidance, controller synthesis, etc., can be turned by means of “Lyapunov theorems” into problems about finding special functions (the *Lyapunov functions*) that satisfy certain sign conditions. The task of constructing Lyapunov functions has traditionally been one of the most fundamental and challenging tasks in control. In recent years, however, advances in convex programming and in particular in the theory of semidefinite optimization have allowed for the search for Lyapunov functions to become *fully automated*. Figure 2 summarizes the steps involved in this process.

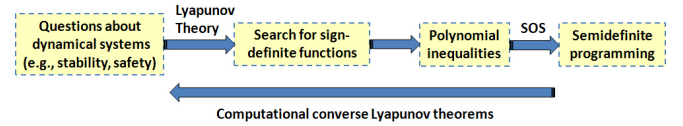


Fig. 2. The steps involved in Lyapunov analysis of dynamical systems via semidefinite programming. The need for “computational” converse Lyapunov theorems is discussed in Section V.

As a simple example, if the task in the leftmost block of Figure 2 is to establish global asymptotic stability of the origin for a polynomial differential equation  $\dot{x} = f(x)$ , with  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $f(0) = 0$ , then the Lyapunov inequalities that a radially unbounded Lyapunov function  $V$  would need to satisfy are [21]:

$$\begin{aligned} V(x) &> 0 && \forall x \neq 0 \\ \dot{V}(x) &= \langle \nabla V(x), f(x) \rangle < 0 && \forall x \neq 0. \end{aligned} \quad (5)$$

Here,  $\dot{V}$  denotes the time derivative of  $V$  along the trajectories of  $\dot{x} = f(x)$ ,  $\nabla V(x)$  is the gradient vector of  $V$ , and  $\langle \cdot, \cdot \rangle$  is the standard inner product in  $\mathbb{R}^n$ . If we parametrize  $V$  as an unknown polynomial function, then the Lyapunov inequalities in (5) become polynomial positivity conditions. The standard sos relaxation for these inequalities would then be:

$$V \text{ sos} \quad \text{and} \quad -\dot{V} = -\langle \nabla V, f \rangle \text{ sos}. \quad (6)$$

The search for a polynomial function  $V$  satisfying these two sos constraints is a semidefinite program, which, if feasible,

would imply<sup>2</sup> a solution to (5) and hence a proof of global asymptotic stability through Lyapunov’s theorem.

#### IV. MORE TRACTABLE ALTERNATIVES TO SUM OF SQUARES OPTIMIZATION [AREA 1]

As explained in Section III, a central question of relevance to applications of algorithmic algebra is to provide sufficient conditions for nonnegativity of polynomials, as working with nonnegativity constraints directly is in general intractable. The sum of squares (sos) condition achieves this goal and is amenable to semidefinite programming (SDP). Although this has proven to be a powerful approach, its application to many practical problems has been challenged by a simple bottleneck: *scalability*.

For a polynomial of degree  $2d$  in  $n$  variables, the size of the semidefinite program that decides the sos decomposition is roughly  $n^d$ . Although this number is polynomial in  $n$  for fixed  $d$ , it can grow rather quickly even for low degree polynomials.

In addition to being large-scale, the resulting semidefinite programs are also often ill-conditioned and challenging to solve. In general, SDPs are among the most expensive convex relaxations and many practitioners try to avoid them when possible. In the field of integer programming for instance, the cutting-plane approaches used on industrial problems are almost exclusively based on linear programming (LP) or second order cone programming (SOCP). Even though semidefinite cuts are known to be stronger, they are typically too expensive to be used even at the root node of branch-and-bound techniques for integer programming. Because of this, many high-performance solvers, e.g., the CPLEX package of IBM [23], do not even provide an SDP solver and instead solely work with LP and SOCP relaxations. In the field of sum of squares optimization, however, a sound alternative to sos programming that can avoid SDP and take advantage of the existing mature and high-performance LP/SOCP solvers is lacking. This is precisely what we aim to achieve in this section.

Let  $PSD_{n,d}$  and  $SOS_{n,d}$  respectively denote the cone of nonnegative and sum of squares polynomials in  $n$  variables and degree  $d$ , with the obvious inclusion relation  $SOS_{n,d} \subseteq PSD_{n,d}$ . The basic idea is to approximate the cone  $SOS_{n,d}$  from the inside with new cones that are more tractable for optimization. Towards this goal, one may think of several natural sufficient conditions for a polynomial to be a sum of squares. For example, consider the following sets:

- The cone of polynomials that are sums of 4-th powers of polynomials:  $\{p \mid p = \sum q_i^4\}$ ,
- The set of polynomials that are a sum of three squares of polynomials:  $\{p \mid p = q_1^2 + q_2^2 + q_3^2\}$ .

Even though both of these sets clearly reside inside the sos cone, they are not any easier to optimize over. In fact, they are much harder! Indeed, testing whether a (quartic)

<sup>2</sup>Here, we are assuming a strictly feasible solution to the SDP (which unless the SDP has an empty interior will be automatically returned by the interior point solver). See the discussion in [22, p. 41].

polynomial is a sum of 4-th powers is NP-hard [24] (as the cone of 4-th powers of linear forms is dual to the cone of nonnegative quartic forms [25]) and optimizing over polynomials that are sums of three squares is intractable (as this task even for quadratics subsumes the NP-hard problem of positive semidefinite matrix completion with a rank constraint [26]). These examples illustrate the rather obvious point that inclusion relationship in general has no implications in terms of complexity of optimization. Indeed, we would need to take some care in deciding what subset of  $SOS_{n,d}$  we exactly choose to work with—on one hand, it has to comprise a “big enough” subset to be useful in practice; on the other hand, it should be computationally simpler for optimization.

##### A. The cone of $r$ -dsos and $r$ -sdsos polynomials

We now describe cones inside  $SOS_{n,d}$  (and some incomparable with  $SOS_{n,d}$  but still inside  $PSD_{n,d}$ ) that are naturally motivated and that lend themselves to linear and second order cone programming. There are also several generalizations of these cones, including some that result in fixed-size (and “small”) semidefinite programs. These can be found in [27] and are omitted from here.

*Definition 1 (Ahmadi, Majumdar, ‘13):*

- A polynomial  $p$  is *diagonally-dominant-sum-of-squares* (dsos) if it can be written as

$$p = \sum_i \alpha_i m_i^2 + \sum_{i,j} \beta_{ij}^+ (m_i + m_j)^2 + \beta_{ij}^- (m_i - m_j)^2,$$

for some monomials  $m_i, m_j$  and some constants  $\alpha_i, \beta_{ij}^+, \beta_{ij}^- \geq 0$ .

- A polynomial  $p$  is *scaled-diagonally-dominant-sum-of-squares* (sdsos) if it can be written as

$$p = \sum_i \alpha_i m_i^2 + \sum_{i,j} (\beta_i^+ m_i + \gamma_j^+ m_j)^2 + (\beta_i^- m_i - \gamma_j^- m_j)^2,$$

for some monomials  $m_i, m_j$  and some constants  $\alpha_i, \beta_i^+, \gamma_j^+, \beta_i^-, \gamma_j^- \geq 0$ .

- For a positive integer  $r$ , a polynomial  $p$  is  *$r$ -diagonally-dominant-sum-of-squares* ( $r$ -dsos) if  $p \cdot (1 + \sum_i x_i^2)^r$  is dsos.

- For a positive integer  $r$ , a polynomial  $p$  is  *$r$ -scaled-diagonally-dominant-sum-of-squares* ( $r$ -sdsos) if  $p \cdot (1 + \sum_i x_i^2)^r$  is sdsos.

We denote the set of polynomials in  $n$  variables and degree  $d$  that are dsos, sdsos,  $r$ -dsos, and  $r$ -sdsos by  $DSOS_{n,d}, SDSOS_{n,d}, rDSOS_{n,d}, rSDSOS_{n,d}$ , respectively.

The following inclusion relations are straightforward:

$$DSOS_{n,d} \subseteq SDSOS_{n,d} \subseteq SOS_{n,d} \subseteq POS_{n,d},$$

$$rDSOS_{n,d} \subseteq rSDSOS_{n,d} \subseteq POS_{n,d}, \forall r,$$

$$rDSOS_{n,d} \subseteq (r+1)DSOS_{n,d}, \forall r,$$

$$rSDSOS_{n,d} \subseteq (r+1)SDSOS_{n,d}, \forall r.$$

Our terminology in Definition 1 comes from the following concepts in linear algebra.

*Definition 2:* A symmetric matrix  $A$  is *diagonally dominant* (dd) if  $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$  for all  $i$ . A symmetric matrix  $A$  is *scaled diagonally dominant* (sdd) if there exists an element-wise positive vector  $y$  such that:

$$a_{ii}y_i \geq \sum_{j \neq i} |a_{ij}|y_j, \forall i.$$

Equivalently,  $A$  is sdd if there exists a positive diagonal matrix  $D$  such that  $AD$  (or equivalently,  $DAD$ ) is dd. We denote the set of  $n \times n$  dd and sdd matrices with  $DD_n$  and  $SDD_n$  respectively.

*Theorem 4.1 (Ahmadi, Majumdar, '13):*

- A polynomial  $p$  of degree  $2d$  is dsos if and only if it admits a representation as  $p(x) = z^T(x)Qz(x)$ , where  $z(x)$  is the standard monomial vector of degree  $d$ , and  $Q$  is a dd matrix.
- A polynomial  $p$  of degree  $2d$  is sdsos if and only if it admits a representation as  $p(x) = z^T(x)Qz(x)$ , where  $z(x)$  is the standard monomial vector of degree  $d$ , and  $Q$  is a sdd matrix.

*Theorem 4.2 (Ahmadi, Majumdar, '13):* For any nonnegative integer  $r$ , the set  $rDSOS_{n,d}$  is polyhedral and the set  $rSDSOS_{n,d}$  has a second order cone representation. For any fixed  $r$  and  $d$ , optimization over  $rDSOS_{n,d}$  (resp.  $rSDSOS_{n,d}$ ) can be done with linear programming (resp. second order cone programming), of size polynomial in  $n$ .

### B. How fast/powerful is the dsos and sdsos methodology?

A large portion of our recent papers [27], [28], [29] is devoted to this question. We provide a glimpse of the results in this section.

As it is probably obvious, the purpose of the parameter  $r$  in Definition 1 is to have a knob for trading off speed with approximation quality. By increasing  $r$ , we obtain increasingly accurate inner approximations to the set of nonnegative polynomials. The following example shows that even the linear programs obtained from  $r = 1$  can outperform the semidefinite programs resulting from sum of squares.

*Example 4.1:* Consider the polynomial

$$p(x) = x_1^4 x_2^2 + x_2^4 x_3^2 + x_3^4 x_1^2 - 3x_1^2 x_2^2 x_3^2.$$

One can show that this polynomial is nonnegative but *not* a sum of squares [20]. However, we can give an LP-based nonnegativity certificate of this polynomial by showing that  $p \in 1DSOS$ . Hence,  $1DSOS \not\subseteq SOS$ .

By employing appropriate Positivstellensatz results from real algebraic geometry, we can prove that many asymptotic guarantees that hold for sum of squares programming also hold for dsos and sdsos programming.

*Theorem 4.3 (Ahmadi, Majumdar, '13):*

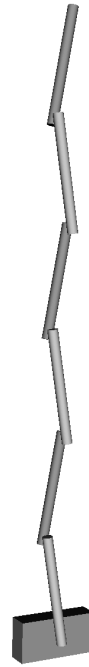


Fig. 3. An illustration of the  $N$ -link inverted pendulum system (with  $N = 6$ ).

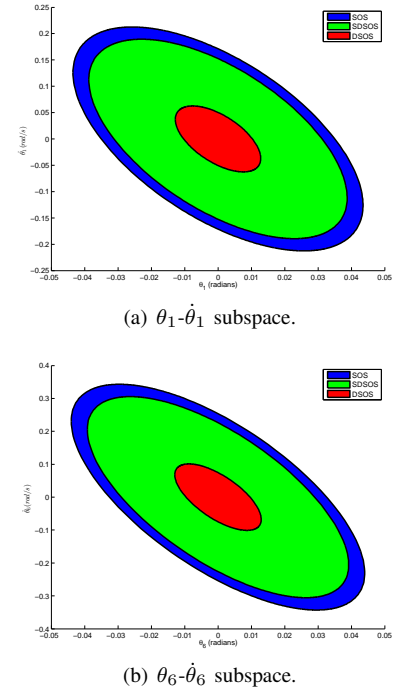


Fig. 4. From [29] (with Majumdar and Tedrake): comparisons of projections of the ROAs computed for the 6-link pendulum system using DSOS, SDSOS and SOS programming, via LP, SOCP, and SDP respectively.

- Let  $p$  be an even form (i.e., a form where no variable is raised to an odd power). If  $p(x) > 0$  for all  $x \neq 0$ , then there exists an integer  $r$  such that  $p \in rDSOS$ .
- Let  $p$  be any form. If  $p(x) > 0$  for all  $x \neq 0$ , then there exists a form  $q$  such that  $q$  is dsos and  $pq$  is dsos. (Observe that this is a certificate of nonnegativity of  $p$  that can be found with linear programming.)
- Consider the polynomial optimization problem (POP) (3) and the hierarchy of sum of squares relaxations of Parrilo [18] that solve it to arbitrary accuracy. If one replaces all sos conditions in this hierarchy with dsos conditions, one still solves POP to arbitrary accuracy (but with a sequence of linear programs instead of semidefinite programs).

On the practical side, we have preliminary evidence for major speed-ups with minor sacrifices in conservatism. Figure 4 shows our experiments for computing the region of attraction (ROA) for the upright equilibrium point of a stabilized inverted  $N$ -link pendulum with  $2N$  states; see Figure 3 for an illustration with  $N = 6$  and [29] for experiments with other values of  $N$ . The same exact algorithm was run (details are in [29]), but polynomials involved in the optimization which were required to be sos, were instead required to be dsos and sdsos. Even the dsos program here is able to do a good job at stabilization. More impressively, the volume of the ROA of the sdsos program is 79% of that of the sos program. For this problem, the speed up of the dsos and sdsos algorithms over the sos algorithm is roughly a factor of 1400 (when SeDuMi is used to solve the SDP) and a factor of 90 (when Mosek is used to solve the SDP).



Perhaps more important than the ability to achieve speedups over the sos approach in small or medium sized problems is the opportunity to work in much bigger regimes where sos solvers have no chance of getting past even the first iteration of the interior point algorithm (at least with the current state of affairs). For example, in work with Majumdar and Tedrake [29], we use sdsos optimization to compute (in the order of minutes) a stabilizing controller and a region of attraction for an equilibrium point of a nonlinear model of the ATLAS robot (built by Boston Dynamics Inc. and used for the 2013 DARPA Robotics Challenge), which has 30 states and 14 control inputs. (See video made by Majumdar and Tedrake: <https://www.youtube.com/watch?v=6jhCiuQVOaQ>) Similarly, in [27], we have been able to solve dense polynomial optimization problems of degree 4 in 70 variables in a few minutes.

**Opportunities for future research.** We believe the most exciting opportunity for new contributions here is to reveal novel application areas in control and polynomial optimization where problems have around 20 – 100 state variables and can benefit from tools for optimization over nonnegative polynomials. It would be interesting to see for which applications, and to what extent, our new dsos and sdsos optimization tools can fill the gap for sos optimization at this scale. To ease such investigations, a MATLAB package for dsos and sdsos optimization is soon to be released as part of the SPOTless toolbox<sup>3</sup>.

On the theoretical side, comparing worst-case approximation guarantees of dsos, sdsos, and sos approaches for particular classes of polynomial optimization problems (beyond our asymptotic results) remains a wide open area.

## V. COMPUTATIONAL ADVANCES IN LYAPUNOV THEORY [AREAS 1&2]

If we place the theory of dynamical systems under a computational lens, our understanding of the theory of nonlinear or hybrid systems is seen to be very primitive compared to that of linear systems. For linear systems, most properties of interest (e.g., stability, boundedness of trajectories, etc.) can be decided in polynomial time. Moreover, there are certificates for all of these properties in form of Lyapunov functions that are *quadratic*. Quadratic functions are tractable for optimization purposes. By contrast, there is no such theory for nonlinear systems. Even for the class of polynomial differential equations of degree two, we do not currently know whether there is a *finite time* (let alone polynomial time) algorithm that can decide stability. In fact, a well-known conjecture of Arnold from [30] states that there should not be such an algorithm. Likewise, the classical converse Lyapunov theorems that we have only guarantee existence of Lyapunov functions within very broad classes of functions (e.g. the class of continuously differentiable functions) that are a priori not amenable to computation. The situation for hybrid systems is similar, if not worse.

We have spent some of our recent research efforts [31], [32], [33], [34], [35] understanding the behavior of nonlinear (mainly polynomial) and hybrid (mainly switched linear) dynamical systems both in terms of computational complexity and existence of computationally friendly Lyapunov functions. In a nutshell, the goal has been to establish results along the “converse arrow” of Figure 2 in Section III. Some of our results are encouraging. For example, we have shown that under certain conditions, existence of a polynomial Lyapunov function for a polynomial differential equation implies existence of a Lyapunov function that can be found with sum of squares techniques and semidefinite programming [31], [33]. More recently, we have shown that stability of switched linear systems implies existence of an *sos-convex* Lyapunov functions [36]. These are Lyapunov functions that can be found with semidefinite programming and that have algebraic certificates of convexity [36], [37]. Unfortunately, however, we also have results that are very negative in nature:

*Theorem 5.1 (Ahmadi, Krstic, Parrilo [32]):* The quadratic polynomial vector field,

$$\begin{aligned}\dot{x} &= -x + xy \\ \dot{y} &= -y,\end{aligned}\tag{7}$$

is globally asymptotically stable but does not admit a polynomial Lyapunov function of any degree.

*Theorem 5.2 (Ahmadi, Parrilo [33]):* For any positive integer  $d$ , there exist *homogeneous*<sup>4</sup> polynomial vector fields in 2 variables and degree 3 that are globally asymptotically stable but do not admit a polynomial Lyapunov function of degree  $\leq d$ .

*Theorem 5.3 (Ahmadi, Jungers [38]):* Consider the switched linear system  $x_{k+1} = A_i x_k$ . For any positive integer  $d$ , there exist pairs of  $2 \times 2$  matrices  $A_1, A_2$  that are asymptotically stable under arbitrary switching but do not admit (i) a polynomial Lyapunov function of degree  $\leq d$ , or (ii) a polytopic Lyapunov function with  $\leq d$  facets, or (iii) a piecewise quadratic Lyapunov function with  $\leq d$  pieces. (This implies that there cannot be an upper bound on the size of the linear and semidefinite programs that search for such stability certificates.)

*Theorem 5.4 (Ahmadi [34]):* Unless P=NP, there cannot be a polynomial time (or even pseudo-polynomial time) algorithm for deciding whether the origin of a cubic polynomial differential equation is locally (or globally) asymptotically stable.

*Theorem 5.5 (Ahmadi, Majumdar, Tedrake [35]):* The hardness result of Theorem 5.4 extends to ten other fundamental properties of polynomial differential equations such as boundedness of trajectories, invariance of sets, stability in the sense of Lyapunov, collision avoidance, stabilizability by linear feedback, and others.

These results show a sharp transition in complexity of Lyapunov functions when we move away from linear systems ever so slightly. Although one may think that such

<sup>3</sup><https://github.com/spot-toolbox/spotless>

<sup>4</sup>A homogeneous polynomial vector field is one where all monomials have the same degree. Linear systems are an example.

counterexamples are not representative of the general case, in fact it is quite common for simple nonlinear or hybrid dynamical systems to at least necessitate “complicated” (e.g., high degree) Lyapunov functions. In view of this, it is natural to ask whether we can replace the standard Lyapunov inequalities with new ones that are less stringent in their requirements but still imply stability. This would enlarge the class of valid stability certificates to include simpler functions and hence reduce the size of the optimization problems that try to construct these functions.

In this direction, we have developed two frameworks: *path-complete graph Lyapunov functions* (with Jungers and Roozbehani) [39], [40] and *non-monotonic Lyapunov functions* [22], [41]. The first approach is based on the idea of using *multiple* Lyapunov functions instead of one and brings in concepts from automata theory to establish how Lyapunov inequalities should be written among multiple Lyapunov functions. The second approach relaxes the classical requirement that Lyapunov functions should *monotonically* decrease along trajectories. We briefly describe these concepts next.

#### A. Lyapunov inequalities and transitions in finite automata [Areas 1&2]

Consider a finite set of matrices  $\mathcal{A} := \{A_1, \dots, A_m\}$ . Our goal is to establish *global asymptotic stability under arbitrary switching* (GASUAS) of the difference inclusion system

$$x_{k+1} \in \text{co}\mathcal{A} x_k, \quad (8)$$

where  $\text{co}\mathcal{A}$  here denotes the convex hull of the set  $\mathcal{A}$ . In other words, we would like to prove that no matter what the realization of our *uncertain and time-varying* linear system turns out to be at each time step, as long as it stays within  $\text{co}\mathcal{A}$ , then we have stability. Let  $\rho(\mathcal{A})$  be the *joint spectral radius* (JSR) of the set of matrices  $\mathcal{A}$ :

$$\rho(\mathcal{A}) = \lim_{k \rightarrow \infty} \max_{\sigma \in \{1, \dots, m\}^k} \|A_{\sigma_k} \dots A_{\sigma_2} A_{\sigma_1}\|^{1/k}. \quad (9)$$

It is well-known that  $\rho < 1$  if and only if system (8) is GASUAS.

Aside from stability of switched systems, computation of the JSR emerges in many areas of application such as computation of the capacity of codes, continuity of wavelet functions, convergence of consensus algorithms, trackability of graphs, and many others; see [42]. In [39], [40], we give SDP-based approximation algorithms for the JSR by applying Lyapunov analysis techniques to system (8). We show that considerable improvements in scalability are possible (especially for high dimensional systems) if instead of a common Lyapunov function of high degree for the set  $\mathcal{A}$ , we use *multiple* Lyapunov functions of low degree (quadratic ones). Motivated by this observation, the main challenge is to *understand which sets of inequalities among a finite set of Lyapunov functions imply stability*. We give a graph theoretic answer to this question by defining directed graphs whose nodes are Lyapunov functions and whose edges are labeled with matrices from the set of input matrices  $\mathcal{A}$ . Each edge of

this graph defines a single Lyapunov inequality as depicted in Figure 5(a).

*Definition 3:* (Ahmadi, Jungers, Parrilo, Roozbehani [39]) Given a directed graph  $G(N, E)$  whose edges are labeled with words (matrices) from the set  $\mathcal{A}$ , we say that the graph is *path-complete*, if for all finite words  $A_{\sigma_k} \dots A_{\sigma_1}$  of any length  $k$  (i.e., for all words in  $\mathcal{A}^*$ ), there is a directed path in the graph such that the labels on the edges of this path are the labels  $A_{\sigma_1}$  up to  $A_{\sigma_k}$ .

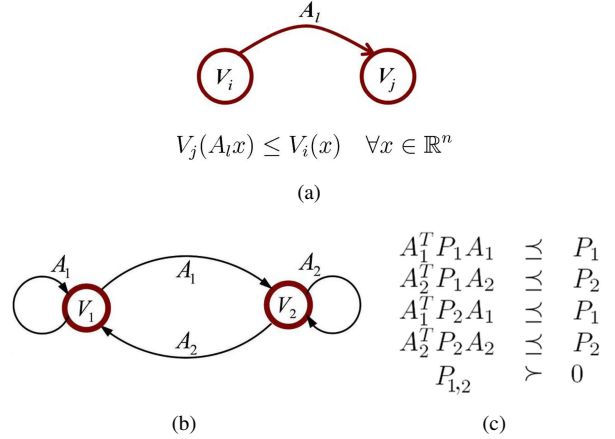


Fig. 5. Path-complete graph Lyapunov functions. (a) The nodes of the graph are Lyapunov functions and the directed edges, which are labeled with matrices from the set  $\mathcal{A}$ , represent Lyapunov inequalities. (b) An example of a path-complete graph on the alphabet  $\{A_1, A_2\}$ . This graph contains a directed path for every finite word. (c) The SDP associated with the graph in (b) when quadratic Lyapunov functions  $V_{1,2}(x) = x^T P_{1,2} x$  are assigned to its nodes. This is an SDP in matrix variables  $P_1$  and  $P_2$  which if feasible implies  $\rho(A_1, A_2) \leq 1$ . We prove an approximation ratio of  $1/\sqrt[4]{n}$  for this particular SDP.

An example of a path-complete graph is given in Figure 5(b), with dozens more given in [40]. In the terminology of automata theory, path-complete graphs correspond precisely to finite automata whose language is the set  $\mathcal{A}^*$  of all words (i.e., matrix products) from the alphabet  $\mathcal{A}$ . There are well-known algorithms in automata theory (see e.g. [43, Chap. 4]) that can check whether the language accepted by an automaton is  $\mathcal{A}^*$ . Similar algorithms exist in the symbolic dynamics literature; see e.g. [44, Chap. 3]. Our interest in path-complete graphs stems from the following two theorems that relate this notion to Lyapunov stability.

*Theorem 5.6:* (Ahmadi, Jungers, Parrilo, Roozbehani [39]) Consider *any* path-complete graph with edges labeled with matrices from the set  $\mathcal{A}$ . Define a set of Lyapunov inequalities, one per edge of the graph, following the rule in Figure 5(a). If Lyapunov functions are found, one per node, that satisfy this set of inequalities, then the switched system in (8) is GASUAS.

*Theorem 5.7:* (Jungers, Ahmadi, Parrilo, Roozbehani [45]) Consider any set of inequalities of the form  $V_j(A_k x) \leq V_i(x)$  among a finite number of Lyapunov functions that imply GASUAS of system (8). Then the graph associated with these inequalities, drawn according to

the rule in Figure 5(a), is necessarily path-complete.

These two theorems together give a characterization of *all stability proving Lyapunov inequalities*. Our result has unified several works in the literature, as we observed that many LMIs that appear in the literature [46], [47], [48], [49], [50], [51], [52] correspond to particular families of path-complete graphs. In addition, the framework has introduced several *new* ways of proving stability with new computational benefits. Finally, by relying on some results in convex geometry, we have been able to prove approximation guarantees (converse results) for the SDPs that search for Lyapunov functions on nodes of path-complete graphs. For example, the upper bound  $\hat{\rho}$  that the SDP in Figure 5(c) produces on the JSR satisfies

$$\frac{1}{\sqrt[n]{n}} \hat{\rho}(\mathcal{A}) \leq \rho(\mathcal{A}) \leq \hat{\rho}(\mathcal{A}).$$

### B. Non-monotonic Lyapunov functions [Area 1]

Our research on this topic is motivated by a very natural question: If all we need for the conclusion of Lyapunov's stability theorem to hold is for the value of the Lyapunov function to *eventually reach zero*, why should we require the Lyapunov function to decrease monotonically? Can we write down conditions that allow Lyapunov functions to increase occasionally, but still guarantee their convergence to zero in the limit? In [22], [41], we showed that this is indeed possible. The main idea is to invoke higher order derivatives of Lyapunov functions (or higher order differences in discrete time). Intuitively, whenever we allow  $\dot{V} > 0$  (i.e.,  $V$  increasing), we should make sure some higher order derivatives of  $V$  are negative, so the rate at which  $V$  increases decreases fast enough for  $V$  to be forced to decrease later in the future. An example of such an inequality for a continuous time dynamical system  $\dot{x} = f(x)$  is [53]:

$$\tau_2 \ddot{V}(x) + \tau_1 \dot{V}(x) + \dot{V}(x) < 0. \quad (10)$$

Here,  $\tau_1$  and  $\tau_2$  are nonnegative constants and by the first three derivatives of the Lyapunov function  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  in this expression, we mean

$$\dot{V}(x) = \left\langle \frac{\partial V(x)}{\partial x}, f(x) \right\rangle,$$

$$\dot{V}(x) = \left\langle \frac{\partial \dot{V}(x)}{\partial x}, f(x) \right\rangle,$$

$$\ddot{V}(x) = \left\langle \frac{\partial \ddot{V}(x)}{\partial x}, f(x) \right\rangle.$$

In [22], [54], we establish a link between non-monotonic Lyapunov functions and standard ones, showing how the latter can be constructed from the former. The main advantage of non-monotonic Lyapunov functions over standard ones is, however, that they can often be much simpler in structure. Figure 6 shows a trajectory of a stable linear time-varying system for which a standard Lyapunov function should either depend on time or be extremely complicated. However, if one uses condition (10), the simple quadratic non-monotonic Lyapunov function  $\|x\|^2$  provides a proof of stability. We have also showed how one can replace

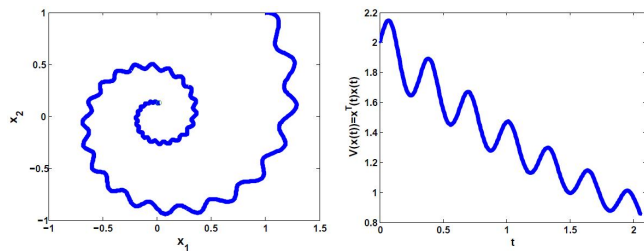


Fig. 6. *Non-monotonic Lyapunov functions* [22]. A typical trajectory of a linear time-varying dynamical system (left). The value of a stability proving quadratic non-monotonic Lyapunov function along the trajectory (right). A classical time-independent Lyapunov function would have to be extremely complicated.

condition (10) with other inequalities involving the first three derivatives, which are at least as powerful, but also *convex* in the decision variables. This allowed for sum of squares methods to become applicable for an automated search for non-monotonic Lyapunov functions. The concrete advantage over standard Lyapunov functions is savings in the number of decision variables of the sos programs; see, e.g., [54, Ex. 2.1].

**Opportunities for future research.** The body of work described in this section leaves several directions for future research:

- On the topic of complexity: What is the complexity of testing asymptotic stability of a polynomial vector field of degree 2? For degree 1, the problem can be solved in polynomial time; for degree 3, we have shown that the problem is strongly NP-hard [34], [33].
- On the topic of existence of polynomial Lyapunov functions: Is there a locally asymptotically stable polynomial vector field with rational coefficients that does not admit a local polynomial Lyapunov function? Our work in [32] presents an example with no *global* polynomial Lyapunov function. Bacciotti and Rosier [55, Prop. 5.2] present an independent example with no local polynomial Lyapunov function, but their vector field needs to have an *irrational* coefficient and the non-existence of polynomial Lyapunov functions for their example is not robust to arbitrarily small perturbations.
- On the topic of existence of sos Lyapunov functions: Does existence of a polynomial Lyapunov function for a polynomial vector field imply existence of an sos Lyapunov function (see [31] for a precise definition)? We have answered this question in the affirmative under a few assumptions [31], [13], but not in general.
- On the topic of path-complete graph Lyapunov functions: Characterize all Lyapunov inequalities among multiple Lyapunov functions that establish switched stability of a *nonlinear* difference inclusion. We know already that the situation is more delicate here than the characterization for the linear case presented in [40]. Indeed, we have shown [36] that path-complete graphs no longer guarantee stability and that *convexity* of Lyapunov functions plays a role in the nonlinear case.



- On the topic of non-monotonic Lyapunov functions: Characterize *all* Lyapunov inequalities involving a finite number of higher order derivatives that imply stability. Determine whether the search for Lyapunov functions satisfying these inequalities can be cast as a convex program.

## VI. ACKNOWLEDGEMENTS

We are grateful to Anirudha Majumdar for his contributions to the work presented in Section IV and to Russ Tedrake for the robotics applications and many insightful discussions.

## REFERENCES

- [1] N. Gvozdenović and M. Laurent. Semidefinite bounds for the stability number of a graph via sums of squares of polynomials. *Mathematical Programming*, 110(1):145–173, 2007.
- [2] P. A. Parrilo. Polynomial games and sum of squares optimization. In *Proceedings of the 45<sup>th</sup> IEEE Conference on Decision and Control*, 2006.
- [3] Dimitris Bertsimas, Dan Andrei Iancu, and Pablo A Parrilo. A hierarchy of near-optimal policies for multistage adaptive optimization. *IEEE Transactions on Automatic Control*, 56(12):2809–2824, 2011.
- [4] A. Magnani, S. Lall, and S. Boyd. Tractable fitting with convex polynomials via sum of squares. In *Proceedings of the 44<sup>th</sup> IEEE Conference on Decision and Control*, 2005.
- [5] M. Roozbehani. *Optimization of Lyapunov invariants in analysis and implementation of safety-critical software systems*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [6] Roh Tae, Bogdan Dumitrescu, and Lieven Vandenberghe. Multidimensional FIR filter design via trigonometric sum-of-squares optimization. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):641–650, 2007.
- [7] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri. Distinguishing separable and entangled states. *Physical Review Letters*, 88(18), 2002.
- [8] John Harrison. Verifying nonlinear real formulas via sums of squares. In *Theorem Proving in Higher Order Logics*, pages 102–118. Springer, 2007.
- [9] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, May 2000.
- [10] D. Henrion and A. Garulli, editors. *Positive polynomials in control*, volume 312 of *Lecture Notes in Control and Information Sciences*. Springer, 2005.
- [11] G. Chesi and D. Henrion (editors). Special issue on positive polynomials in control. *IEEE Trans. Automat. Control*, 54(5), 2009.
- [12] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares programming. In *Proceedings of the 42<sup>th</sup> IEEE Conference on Decision and Control*, pages 4676–4681, 2003.
- [13] A. A. Ahmadi. *Algebraic relaxations and hardness results in polynomial optimization and Lyapunov analysis*. PhD thesis, Massachusetts Institute of Technology, September 2011. Available at <http://aaa.lids.mit.edu/publications>.
- [14] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control design along trajectories with sums of squares programming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.
- [15] Abhijit Chakraborty, Peter Seiler, and Gary J Balas. Susceptibility of F/A-18 flight controllers to the falling-leaf mode: Nonlinear analysis. *Journal of guidance, control, and dynamics*, 34(1):73–85, 2011.
- [16] Peter Seiler, Gary J Balas, and Andrew K Packard. Assessment of aircraft flight controllers using nonlinear robustness analysis techniques. In *Optimization Based Clearance of Flight Control Laws*, pages 369–397. Springer, 2012.
- [17] A. Ataei-Esfahani and Q. Wang. Nonlinear control design of a hypersonic aircraft using sum-of-squares methods. In *Proceedings of the American Control Conference*, pages 5278–5283. IEEE, 2007.
- [18] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2, Ser. B):293–320, 2003.
- [19] D. Hilbert. Über die Darstellung Definitiver Formen als Summe von Formenquadraten. *Math. Ann.*, 32, 1888.
- [20] B. Reznick. Some concrete aspects of Hilbert’s 17th problem. In *Contemporary Mathematics*, volume 253, pages 251–272. American Mathematical Society, 2000.
- [21] H. Khalil. *Nonlinear systems*. Prentice Hall, 2002. Third edition.
- [22] A. A. Ahmadi. Non-monotonic Lyapunov functions for stability of nonlinear and switched systems: theory and computation. Master’s thesis, Massachusetts Institute of Technology, June 2008. Available at <http://aaa.lids.mit.edu/publications>.
- [23] CPLEX. V12. 2: Users manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2010.
- [24] Christopher Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *arXiv preprint arXiv:0911.1393*, 2009.
- [25] B. Reznick. Uniform denominators in Hilbert’s 17th problem. *Math Z.*, 220(1):75–97, 1995.
- [26] E. Marianna, M. Laurent, and A. Varvitsiotis. Complexity of the positive semidefinite matrix completion problem with a rank constraint. In *Discrete Geometry and Optimization*, pages 105–120. Springer, 2013.
- [27] A. A. Ahmadi and A. Majumdar. DSOS and SDSOS: more tractable alternatives to sum of squares and semidefinite optimization. In preparation, 2014.
- [28] A. A. Ahmadi and A. Majumdar. Some applications of polynomial optimization in operations research and real-time decision making. Under review, 2014.
- [29] A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control and verification of high-dimensional systems via dsos and sdsos optimization. Submitted to the 53rd IEEE conference on decision and control, 2014.
- [30] V. I. Arnold. Problems of present day mathematics, XVII (Dynamical systems and differential equations). *Proc. Symp. Pure Math.*, 28(59), 1976.
- [31] A. A. Ahmadi and P. A. Parrilo. Converse results on existence of sum of squares Lyapunov functions. In *Proceedings of the 50<sup>th</sup> IEEE Conference on Decision and Control*, 2011.
- [32] A. A. Ahmadi, M. Krstic, and P. A. Parrilo. A globally asymptotically stable polynomial vector field with no polynomial Lyapunov function. In *Proceedings of the 50<sup>th</sup> IEEE Conference on Decision and Control*, 2011.
- [33] A. A. Ahmadi and P. A. Parrilo. Stability of polynomial differential equations: complexity and converse Lyapunov questions. Under review. Preprint available at <http://arxiv.org/abs/1308.6833>, 2013.
- [34] A. A. Ahmadi. On the difficulty of deciding asymptotic stability of cubic homogeneous vector fields. In *Proceedings of the American Control Conference*, 2012.
- [35] A. A. Ahmadi, A. Majumdar, and R. Tedrake. Complexity of ten decision problems in continuous time dynamical systems. In *Proceedings of the American Control Conference*, 2013.
- [36] A. A. Ahmadi and R. Jungers. SOS-convex Lyapunov functions with applications to nonlinear switched systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2013.
- [37] A. A. Ahmadi and P. A. Parrilo. A complete characterization of the gap between convexity and sos-convexity. *SIAM Journal on Optimization*, 23(2):811–833, 2013.
- [38] A. A. Ahmadi and R. Jungers. On complexity of Lyapunov functions for switched linear systems. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, 2014.
- [39] A. A. Ahmadi, R. Jungers, P. A. Parrilo, and M. Roozbehani. Analysis of the joint spectral radius via Lyapunov functions on path-complete graphs. In *Hybrid Systems: Computation and Control 2011*, Lecture Notes in Computer Science. Springer, 2011.
- [40] A. A. Ahmadi, R. Jungers, P. A. Parrilo, and M. Roozbehani. Joint spectral radius and path-complete graph Lyapunov functions. *SIAM Journal on Optimization and Control*, 52(1), 2014.
- [41] A. A. Ahmadi and P. A. Parrilo. Non-monotonic Lyapunov functions for stability of discrete time nonlinear and switched systems. In *Proceedings of the 47<sup>th</sup> IEEE Conference on Decision and Control*, 2008.
- [42] R. Jungers. *The joint spectral radius: theory and applications*, volume 385 of *Lecture Notes in Control and Information Sciences*. Springer, 2009.
- [43] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.
- [44] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

- [45] A. A. Ahmadi, R. M. Jungers, P. A. Parrilo, and M. Roozbehani. When is a set of LMIs a sufficient condition for stability? In *Proceedings of the IFAC Symposium on Robust Control Design*, 2012.
- [46] T. Hu, L. Ma, and Z. Li. On several composite quadratic Lyapunov functions for switched systems. In *Proceedings of the 45<sup>th</sup> IEEE Conference on Decision and Control*, 2006.
- [47] T. Hu and Z. Lin. Absolute stability analysis of discrete-time systems with composite quadratic Lyapunov functions. *IEEE Transactions on Automatic Control*, 50(6):781–797, 2005.
- [48] R. Goebel, A. R. Teel, T. Hu, and Z. Lin. Conjugate convex Lyapunov functions for dual linear differential inclusions. *IEEE Transactions on Automatic Control*, 51(4):661–666, 2006.
- [49] T. Hu, L. Ma, and Z. Lin. Stabilization of switched systems via composite quadratic functions. *IEEE Transactions on Automatic Control*, 53(11):2571 – 2585, 2008.
- [50] J. W. Lee and G. E. Dullerud. Uniform stabilization of discrete-time switched and Markovian jump linear systems. *Automatica*, 42(2):205–218, 2006.
- [51] J. Daafouz and J. Bernussou. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems and Control Letters*, 43(5):355–359, 2001.
- [52] J. W. Lee and P. P. Khargonekar. Detectability and stabilizability of discrete-time switched linear systems. *IEEE Transactions on Automatic Control*, 54(3):424–437, 2009.
- [53] A. R. Butz. Higher order derivatives of Liapunov functions. *IEEE Trans. Automatic Control*, AC-14:111–112, 1969.
- [54] A. A. Ahmadi and P. A. Parrilo. On higher order derivatives of Lyapunov functions. In *Proceedings of the 2011 American Control Conference*, 2011.
- [55] A. Bacciotti and L. Rosier. *Liapunov Functions and Stability in Control Theory*. Springer, 2005.