A Tutorial on Sum of Squares Techniques for Systems Analysis

Antonis Papachristodoulou and Stephen Prajna

***Abstract*— This tutorial is about new system analysis techniques that were developed in the past few years based on the sum of squares decomposition. We will present stability and robust stability analysis tools for different classes of systems: systems described by nonlinear ordinary differential equations or differential algebraic equations, hybrid systems with nonlinear subsystems and/or nonlinear switching surfaces, and time-delay systems described by nonlinear functional differential equations. We will also discuss how different analysis questions such as model validation and safety verification can be answered for uncertain nonlinear and hybrid systems.**

I. Introduction

The sum of squares (SOS) technique was introduced nearly five years ago in Parrilo's thesis [34]. Ever since a plethora of questions on systems analysis were tackled, that were difficult to answer before. For example, it allowed the algorithmic analysis of nonlinear systems using Lyapunov methods — upon which most of nonlinear systems theory is based. Apart from stability analysis of nonlinear systems, the SOS technique has opened a new direction in the treatment of other system types and answering different analysis and synthesis questions.

The SOS approach generalizes a well known algorithmic tool in linear robust control theory — linear matrix inequalities (LMIs) [5]. Using LMIs to formulate several analysis and synthesis problems is advantageous, as there are efficient algorithms to solve them, developed in the framework of semidefinite programming (SDP) [55] with a complexity that is worst-case polynomial in time. The SOS technique uses exactly the same algorithm, but all questions are formulated at the polynomial or polynomial matrix level.

Take as an example the problem of proving stability for an equilibrium of a dynamical system $\dot{x} = f(x)$ using what is called a *Lyapunov function* [19]. All that is required to prove stability is to find a positive definite function $V(x)$ defined in some region of the state space containing the equilibrium point whose derivative $\dot{V}(x) = \frac{\partial V}{\partial x} f(x)$ is negative semidefinite along the system trajectories. In the linear case, these conditions amount to finding, for a system $\dot{x} = Ax$, a positive definite matrix P such that $A^T P + PA$ is negative definite [5]; then the associated Lyapunov function is given by $V(x) = x^T P x$. What is obvious but was not seen from that angle until recently, is that both $V(x)$ and $-\dot{V}(x)$ are also *sums of squares*.

Hand-crafted Lyapunov functions constructions are inevitably limited to small state dimensions and depend on the analytic skills of the researcher. In the case in which both the vector field f and the Lyapunov function candidate V are polynomial, the Lyapunov conditions are essentially polynomial non-negativity conditions which can be NP hard to test [28] — probably one of the reasons for the lack of algorithmic constructions of Lyapunov functions. However, if we replace the nonnegativity conditions by SOS conditions, then not only testing the Lyapunov function conditions — but also constructing the Lyapunov function — can be done efficiently using semidefinite programming (SDP) [55]. This observation can be used to answer other analysis questions for more complicated system descriptions algorithmically.

This tutorial begins with an overview of the theory and computational aspects of the SOS decomposition. To facilitate the transformation of the polynomial formulation to the corresponding SDP formulation — that is cumbersome if performed manually — software has been developed. One such software is SOSTOOLS [44], a freely-available MATLAB toolbox for solving SOS programs. A brief overview of SOSTOOLS will be given in Section III.

The SOS methodology has allowed the stability analysis of other system classes, including constrained nonlinear, hybrid, and time-delay, as well as the verification of other properties for these systems, such as model invalidation or safety verification. The rest of this tutorial describes these advances, through illustrative examples.

The first class of systems we will be looking at is nonlinear constrained systems. We present an extension to Lyapunov stability theory to allow for the algorithmic analysis of systems that evolve over equality, inequality and integral constraints [32]. A special case of this

Work financially supported by AFOSR MURI, NIH/NIGMS AfCS (Alliance for Cellular Signalling), DARPA, the Kitano ERATO Systems Biology Project, and URI "Protecting Infrastructures from Themselves".

A. Papachristodoulou and S. Prajna are with Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA. Email: {antonis,prajna}@cds.caltech.edu.

This is an overview of work that both authors have developed either together or on their own.

framework is systems with non-polynomial vector fields, which can be treated using a systematic recasting procedure [33]. Section IV deals with these developments.

We will next cover the second class of systems, namely hybrid systems. These are systems that consist of subsystems with a switching rule, and have attracted the attention of researchers for many years now. Traditional analysis methods concentrate on the construction of *quadratic* common or multiple Lyapunov functions using LMI techniques [6], but only for systems with linear subsystems and switching surfaces described by hyperplanes. Using the SOS approach, *polynomial* Lyapunov functions can be constructed for systems with nonlinear subsystems and polynomial switching surfaces. A brief description of the SOS methodologies for treating these systems will be given in Section V.

The third class of systems we will be considering is time-delay systems. They are the simplest infinite dimensional systems with a rich literature on their stability and stabilization properties [20]. LMI techniques have allowed the construction of simple Lyapunov certificates for linear systems, that were many times conservative — the complete Lyapunov functional yields infinite dimensional LMI conditions which are difficult to solve algorithmically. SOS techniques have enabled the algorithmic analysis of nonlinear time delay systems through the solution of these infinite dimensional LMIs. The methodology will be presented in Section VI.

Lyapunov functions should be regarded as *proofs/certificates* guaranteeing the stability property. This aspect of Lyapunov methods is important, as analysis questions are answered in a way that no simulation procedure can. Using Lyapunov-like arguments, other questions apart from stability can be answered algorithmically. For example, estimating L_2 gains in nonlinear systems can be done by constructing appropriate storage functions [56], [57].

More recently, questions such as model validation [39] or safety verification [40], [41] were treated using *barrier certificates*, which can also be constructed using SOS techniques. An overview of the techniques based on barrier certificates will be covered in Sections VII and VIII. Safety verification of hybrid systems within this framework has found application on safe-critical systems of industrial interest, such as assuring the safe performance of a life-support system [14].

This tutorial concludes with some remarks in Section IX.

II. The Sum of Squares Decomposition

In this section we give a brief introduction to sum of squares (SOS) polynomials as well as their use, and show how the existence of a SOS decomposition can be verified using semidefinite programming [55]. A more detailed description can be found in [34], [36] and the references therein. A "dual" approach is given in [24]. There is a wealth of literature on SOS and positive polynomials, especially after Hilbert's 17th problem [49] was answered affirmatively by Artin in 1926, a development that formed the foundation of real algebra and real algebraic geometry [48], [4]. The semidefinite programming method for computing the SOS decomposition is based on the Gram matrix method — see [7], [38] for more details.

Definition 1: For $x \in \mathbb{R}^n$, a multivariate polynomial $p(x)$ is a sum of squares (SOS) if there exist some polynomials $f_i(x)$, $i = 1 \dots M$ such that

$$p(x) = \sum_{i=1}^{M} f_i^2(x). \tag{1}$$

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

Proposition 2: A polynomial $p(x)$ of degree $2d$ is a SOS if and only if there exists a positive semidefinite matrix Q and a vector of monomials $Z(x)$ containing monomials in x of degree $\leq d$ such that

$$p = Z(x)^T Q Z(x) \tag{2}$$

Proof: See [34]. ∎

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 (this was first observed by Parrilo in [34]). The following is an example of how this is done.

Example 3: ([34]) Suppose that we want to know whether or not the quartic polynomial in two variables $p(x_1, x_2) = 2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4$ is a SOS. For this purpose, define $Z(x) = [\; x_1^2 \;\; x_2^2 \;\; x_1 x_2 \;]^T$ and consider the following quadratic form:

$$\begin{aligned} p(x_1, x_2) &= 2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4 \\ &= Z(x)^T \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix}}_{Q} Z(x) \\ &= q_{11} x_1^4 + q_{22} x_2^4 + (2q_{12} + q_{33}) x_1^2 x_2^2 \\ &\quad + 2q_{13} x_1^3 x_2 + 2q_{23} x_1 x_2^3, \end{aligned}$$

from which we get the following relations:

$$\begin{aligned} q_{11} &= 2, & q_{22} &= 5, \\ q_{13} &= 1, & q_{23} &= 0, \\ 2q_{12} + q_{33} &= -1. \end{aligned}$$

Now, decomposing $p(x)$ as an SOS amounts to searching for q_{12} and q_{33} satisfying the last equation, such that $Q \geq 0$. For $q_{12} = -3$ and $q_{33} = 5$, the matrix Q will be positive semidefinite and we have

$$Q = L^T L, \text{ where } L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}.$$

This immediately yields the following SOS decomposition:

$$p(x) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2.$$

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 [49] for more details. Nevertheless, checking nonnegativity of $p(x)$ is an NP-hard problem when the degree of $p(x)$ is at least 4 [28], 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 a semidefinite program, 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 [49], [36]. Note that as the degree of $p(x)$ 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.

Besides the above, 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 feasible values of those coefficients such that $p(x)$ is 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 a semidefinite program (these coefficients are themselves decision variables). This observation is crucial in the construction of Lyapunov functions and other S-procedure type multipliers [34].

Using Positivstellensatz (a central theorem in real algebraic geometry [4]) and the SOS decomposition, the S-procedure can be strengthened to yield less conservative conditions. With Positivstellensatz, a hierarchy of polynomial-time computable stronger conditions can be obtained, and each test is *always* at least as powerful as the standard one, and often strictly stronger. We will be using this feature in the sequel. More details can be found in [34].

III. SOSTOOLS

The observation that the SOS decomposition can be computed efficiently using semidefinite programming [34] has introduced the need for developing software that would facilitate the formulation of the semidefinite programs from their SOS equivalents. One such software is SOSTOOLS [43], [44], [45]: a free, third-party MATLAB[1] toolbox for solving SOS programs.

We define a SOS program as the convex optimization problem of the following form:

$$\begin{aligned} &\text{Minimize } \sum_{j=1}^{J} w_j c_j \\ &\text{subject to} \\ &\qquad a_{i,0}(x) + \sum_{j=1}^{J} a_{i,j}(x) c_j \text{ is SOS, for } i = 1, ..., I, \end{aligned} \tag{3}$$

where the c_j's are the scalar real decision variables, the w_j's are some given real numbers, and the $a_{i,j}(x)$ are some given polynomials (with fixed coefficients). See also another equivalent canonical form of SOS programs in [43], [44]. While the conversion from SOS programs to semidefinite programs (SDPs) can be manually performed for small size instances or tailored for specific problem classes, such a conversion can be quite cumbersome 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 (see Figure 1). 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. In this way the details of the reformulation are abstracted from the user, who can work at the polynomial object level. The user interface of SOSTOOLS has been designed to be simple, easy to use, and transparent while keeping a large degree of flexibility. The current version of SOSTOOLS uses either SeDuMi [52] or SDPT3 [53], both of which are free MATLAB add-ons, as the SDP solver.

The polynomial variables in the SOS programs can be defined in SOSTOOLS in two different ways: using the MATLAB Symbolic Math Toolbox or the custom-built polynomial toolbox. The former method provides the user the benefit of making use of all the features in the toolbox, which range from simple arithmetic

[1] A registered trademark of The MathWorks, Inc.

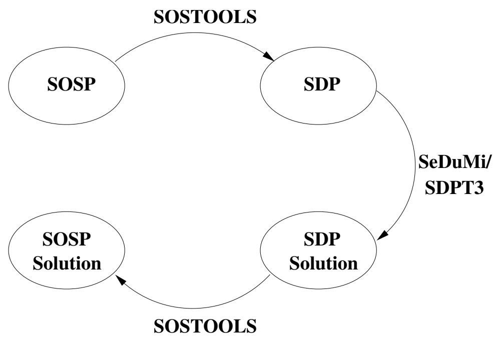

Fig. 1. Diagram depicting how SOS programs (SOSPs) are solved using SOSTOOLS.

operations to differentiation, integration and polynomial manipulation. Even though the integrated polynomial toolbox has only some of the functions of the symbolic toolbox, it allows users that do not have access to the symbolic toolbox to use SOSTOOLS. It also provides an alternative, sometimes faster SDP formulation path.

In many cases the SDPs that we wish to solve have certain structural properties, such as sparsity, symmetry etc. The formulation of the SDP should take them into account: this will not only reduce the computational burden of solving them as their size is many times reduced considerably, but it also removes numerical ill-conditioning. Provision has been taken for sparsity to be taken into account when formulating the SDPs.

The frequent use of certain SOS formulations, such as finding lower bounds on polynomial minima and the search for Lyapunov functions for systems with polynomial vector fields are reflected in the introduction of customized functions in SOSTOOLS. A detailed description of how SOSTOOLS works can be found in the SOSTOOLS user's guide [44].

IV. Analysis of Constrained Systems

We will now turn to systems analysis using SOS techniques. We will first review the available result on constructing Lyapunov functions for systems with polynomial vector fields, and then extend the class of systems for which the construction is possible to nonlinear systems with equality, inequality, and integral constraints. This is a very general class of systems, covering differential algebraic equations and robust stability analysis formulations, areas that have attracted the attention of many researchers in the past.

A. Lyapunov Stability

Here we concentrate on autonomous nonlinear systems of the form

$$\dot{x} = f(x), \tag{4}$$

where $x \in \mathbb{R}^n$ and for which we assume without loss of generality that $f(0) = 0$, i.e. the origin is an equilibrium of the system, and that in a region $\mathcal{D}$ around the origin f is Lipschitz. One of the most important properties related to this equilibrium is its stability, and assessing whether stability of the equilibrium holds can be done by constructing what is called a Lyapunov function.

Theorem 4 ([19]): Consider the system (4), and let $\mathcal{D} \subseteq \mathbb{R}^n$ be a neighborhood of the origin. If there is a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}_+$ such that the following two conditions are satisfied:

1) $V(x) > 0$ for all $x \in \mathcal{D} \setminus \{0\}$ and $V(0) = 0$, i.e., $V(x)$ is positive definite in $\mathcal{D}$
2) $-\dot{V}(x) = -\frac{\partial V}{\partial x} f(x) \geq 0$ for all $x \in \mathcal{D}$, i.e., $\dot{V}(x)$ is negative semidefinite in $\mathcal{D}$

then the origin is a *stable* equilibrium. If in condition (2) above $\dot{V}(x)$ is *negative definite* in $\mathcal{D}$ then the origin is *asymptotically stable*. If $\mathcal{D} = \mathbb{R}^n$ and $V(x)$ is radially unbounded, i.e., $V(x) \to \infty$ as $\|x\| \to \infty$, then the result holds globally.

Let us for now assume that $f(x)$ is a polynomial vector field, and that we will be searching for $V(x)$ that is also a polynomial in x. Then the two conditions in Theorem 4 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 [34]. The only apparent difficulty is the restriction of $V(x)$ to be positive *definite*, not just positive semidefinite. To work around this problem we can use the following proposition.

Proposition 5: Given a polynomial $V(x)$ of degree $2d$, let $\varphi(x) = \sum_{i=1}^{n} \sum_{j=1}^{d} \epsilon_{ij} x_i^{2j}$ such that:

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

with γ a positive number, and $\epsilon_{ij} \geq 0$ for all i and j. Then the condition

$$V(x) - \varphi(x) \text{ is a SOS} \tag{5}$$

guarantees the positive definiteness of $V(x)$.

Proof: The function $\varphi(x)$ as defined above is positive definite if ϵ_{ij}'s satisfy the conditions mentioned in the proposition. Then $V(x) - \varphi(x)$ being SOS implies that $V(x) \geq \varphi(x)$, and therefore $V(x)$ is positive definite. ■

For $\mathcal{D} = \mathbb{R}^n$, the conditions in Theorem 4 can be formulated as SOS conditions as follows:

Proposition 6: Suppose that for the system (4) there exists a polynomial $V(x)$ such that $V(0) = 0$, and

$$V(x) - \varphi(x) \text{ is SOS,} \quad (6)$$

$$-\frac{\partial V}{\partial x} f(x) \text{ is SOS,} \quad (7)$$

where $\varphi(x)$ is as defined in Proposition 5. Then the zero equilibrium of (4) is globally stable (i.e., the equilibrium is stable in the sense of Lyapunov and also all trajectories are bounded).

Below is an example of how the construction of a Lyapunov function is performed using SOSTOOLS.

Example 7: Consider the system

$$\dot{x}_1 = -x_1 + x_2^3 - 3x_3x_4, \qquad \dot{x}_2 = -x_1 - x_2^3,$$
$$\dot{x}_3 = x_1x_4 - x_3, \qquad \dot{x}_4 = x_1x_3 - x_4^3,$$

which has the only equilibrium at the origin. As a first attempt, we will try to construct a quadratic Lyapunov function of the form $V = \sum_{i=1}^{4} \sum_{j=i}^{4} a_{ij}x_ix_j$ where the a_{ij}'s are the unknowns. We search for V that satisfy the conditions in Proposition 6.

It turns out that a Lyapunov function of the above form does not exist (the corresponding semidefinite program is infeasible), so we will next search for a quartic Lyapunov function. One then finds a Lyapunov function that satisfies condition (6) and (7), and thus proves global asymptotic stability of the origin. To 3 significant digits this reads:

$$\begin{aligned} V(x) = {} & 1.12x_1x_2x_3^2 - 0.785x_1x_2 + 0.713x_2^3x_1 \\ & + 0.500x_1x_2x_4^2 + 0.768x_4^4 + 1.64x_1^2 + 1.76x_3^2 \\ & + 0.392x_2^2 + 1.63x_4^2 + 1.69x_1^2x_2^2 + 0.557x_3^4 \\ & + 0.724x_1^3x_2 + 0.181x_1^4 + 1.07x_2^4 + 0.561x_1^2x_3^2 \\ & + 1.61x_2^2x_3^2 + 0.525x_1^2x_4^2 + 0.969x_2^2x_4^2 \\ & + 0.569x_3^2x_4^2 - 0.251x_4x_3x_1 + 0.432x_4x_3x_2. \end{aligned}$$

The above can be obtained by using the `findlyap` command in SOSTOOLS.

B. Stability of Systems with Constraints

In this section we present extensions of Lyapunov's stability theorem for handling systems with equality, inequality and integral quadratic constraints.

Inequality constraints arise naturally when considering *positive systems* or when describing *uncertain parameter sets* for the study of robust stability of systems in the presence of parametric uncertainty.

Equality constraints prove useful when describing systems evolving over a manifold; these are also known as differential algebraic equations or descriptor systems [10]. They also appear in robust stability analysis, as constraints guaranteeing that the equilibrium of the system is at the origin.

The last type of constraints that can be incorporated are integral quadratic constraints (IQCs) [26]. They provide a rich framework to encapsulate many types of uncertainty and unmodelled dynamics: dynamic, time-varying, $\mathcal{L}_2$ bounded uncertainty just to mention a few. Moreover one can formulate performance calculations using IQCs, such as $\mathcal{L}_2$ input-output gain estimation etc.

Consider the nonlinear system

$$\dot{x} \;=\; f(x,u), \quad (8)$$

with the following inequality, equality, and integral constraints that are satisfied by x and u:

$$a_{i_1}(x,u) \le 0, \quad \text{for } i_1 = 1, ..., N_1, \quad (9)$$
$$b_{i_2}(x,u) = 0, \quad \text{for } i_2 = 1, ..., N_2, \quad (10)$$
$$\int_0^T c_{i_3}(x,u)dt \le 0, \text{for } i_3 = 1, ..., N_3, \text{ and } \forall T \ge 0 \quad (11)$$

Here $x \in \mathbb{R}^n$ is the state of the system, and $u \in \mathbb{R}^m$ is a collection of auxiliary variables (such as inputs, non-polynomial functions of states, uncertain parameters, and so on — we will see examples of these in Section IV-C). We assume that $f(x,u)$ has no singularity in $\mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^{m+n}$ is defined as

$$\mathcal{D} = \{(x,u) \in \mathbb{R}^{m+n} \mid a_{i_1}(x,u) \le 0, b_{i_2}(x,u) = 0, \text{ for all } i_1 \text{ and } i_2\}.$$

Without loss of generality, it is also assumed that $f(x,u) = 0$ for $x = 0$ and $u \in \mathcal{D}_u^0$, where

$$\mathcal{D}_u^0 = \{u \in \mathbb{R}^m | (0,u) \in \mathcal{D}\}.$$

The following theorem is an extension of Lyapunov's stability theorem, and can be used to prove that the origin is a stable equilibrium of the above system. It uses a technique reminiscent of the well-known S-procedure [59] in nonlinear and robust control theory.

Theorem 8: Suppose that for the above system there exist functions[2] $V(x)$, $p_{i_1}(x,u)$, $q_{i_2}(x,u)$, and constants $r_{i_3} \ge 0$ such that

- $V(x)$ is positive definite[3] in a neighborhood of the origin.
- $p_{i_1}(x,u) \ge 0$ in $\mathcal{D}$.

Then

$$\begin{aligned} -\frac{\partial V}{\partial x} f(x,u) + \sum p_{i_1}(x,u)a_{i_1}(x,u) \\ + \sum q_{i_2}(x,u)b_{i_2}(x,u) + \sum r_{i_3}c_{i_3}(x,u) \ge 0 \end{aligned} \quad (12)$$

will guarantee that the origin of the state space is a stable equilibrium of the system.

[2]Although not written explicitly here, we assume that we keep track of the indices.

[3]Strictly speaking, it is enough to require V to have a local minimum at the origin.

For a proof see [32].

C. Constrained Systems Analysis: SOS Techniques

In this section we will use Theorem 8 along with the SOS decomposition to analyze various cases of systems with constraints. Algorithmic verification of the conditions in Theorem 8 is difficult unless the conditions are relaxed to SOS conditions. Therefore, in our analysis we use the following assumption and relaxations:

- the vector field $f_x(x,u)$ is assumed to be polynomial or rational, and the constraint functions $a_{i_1}(x,u)$, $b_{i_2}(x,u)$, $c_{i_3}(x,u)$ are assumed to be polynomial.
- we search for bounded degree *polynomial* Lyapunov function V and multipliers p_{i_1}, q_{i_2}.
- polynomial nonnegativity and positive definiteness conditions are relaxed to an SOS condition, and to the condition in Proposition 5, respectively.

For instance, a relaxation of Theorem 8 is stated below.

Proposition 9: Suppose that for the above system there exist polynomial functions $V(x)$, $p_{i_1}(x,u)$, $q_{i_2}(x,u)$, a positive definite function $\varphi(x)$ of the form given in Proposition 5 and constants $r_{i_3} \geq 0$ such that

$$V(x) - \varphi(x) \text{ is SOS,} \tag{13}$$

$$p_{i_1}(x,u) \text{ are SOS for } i_1 = 1,\ldots,N_1, \tag{14}$$

$$-\frac{\partial V}{\partial x} f(x,u) + \sum p_{i_1}(x,u) a_{i_1}(x,u) + \sum q_{i_2}(x,u) b_{i_2}(x,u) + \sum r_{i_3} c_{i_3}(x,u) \text{ is SOS.} \tag{15}$$

Then the origin of the state space is a stable equilibrium of the system.

The polynomials $V(x)$ $p_{i_1}(x,u)$, $q_{i_2}(x,u)$, the constants r_{i_3} and the positive definite function $\varphi(x)$ can be computed using SOSTOOLS [43]. We now turn to a series of applications of the above proposition.

1) Robustness Analysis: In the case of parametric uncertainty, some of the auxiliary variables u may now be taken to be parameters. Additionally, some other auxiliary variables can be used to account for the location of the equilibrium of interest, as for a nonlinear system the location of the equilibrium usually changes when the parameters are varied. The use of equality and inequality constraints in this case is natural: the region of the parameter space that is of interest can be described by inequality constraints, and if the equilibrium moves as the parameters change, one can impose an equality constraint on the corresponding auxiliary variables. Here we present an example motivated by a biological system.

Example 10: Two species models of interacting populations can exhibit limit cycle periodic oscillations [27]. The simplest, but chemically plausible tri-molecular reaction that admits periodic solutions is

$$X \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} A, \quad B \overset{k_2}{\rightarrow} Y, \quad 2X + Y \overset{k_3}{\rightarrow} 3X,$$

in which species X is in dynamical equilibrium with species A with a forward rate of reaction k_1 and a backward rate of reaction k_{-1}, and so on. Using the law of mass action, and non-dimensionalising the equations, we get

$$\dot{u} = a - u + u^2 v, \tag{16}$$

$$\dot{v} = b - u^2 v, \tag{17}$$

where u, v are the non-dimensional concentrations of X and Y, and a, b are non-negative constant parameters that depend on the concentrations of A and B. It is known that for a and b satisfying

$$(b-a) \geq (b+a)^3,$$

the system exhibits a stable limit cycle and the equilibrium point is unstable (see Figure 2).

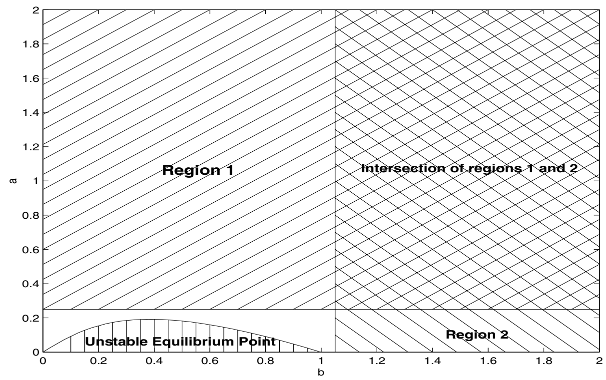

Fig. 2. Chemical Oscillator Example: Stability region for the chemical oscillator problem, Regions 1 and 2 defined by equations (22) and (23).

A Lyapunov function will be constructed for a region of the rest of the parameter space, to prove robust stability of the equilibrium.

The equilibrium of the above system is a ($\bar{u}$, $\bar{v}$) pair that satisfies

$$0 = a - \bar{u} + \bar{u}^2 \bar{v}, \tag{18}$$

$$0 = b - \bar{u}^2 \bar{v}. \tag{19}$$

We translate the equilibrium to the origin using a state transformation $u \to x_1$, $v \to x_2$, $x_1 = u - \bar{u}$, $x_2 = v - \bar{v}$, to get an equivalent system

$$\dot{x}_1 = a - (x_1 + \bar{u}) + (x_1 + \bar{u})^2 (x_2 + \bar{v}), \tag{20}$$

$$\dot{x}_2 = b - (x_1 + \bar{u})^2 (x_2 + \bar{v}), \tag{21}$$

whose equilibrium is at the origin. Now suppose that the parameters a and b are not exactly known, but belong to the set $\{a \geq \underline{a}, b \geq \underline{b}\}$. Notice that when the parameters a and b are changed, the equilibrium $(\bar{u}, \bar{v})$ also changes, see Equations (18)–(19). So there are four parameters in the state equations (20)–(21) - namely a, b, $\bar{u}$, and $\bar{v}$ - that are coupled via the algebraic equality constraints (18)–(19). Denote these parameters (which can be regarded as auxiliary variables) by u_1 through u_4, in accordance with the notation in Theorem 8.

There exist inherent constraints on the state variables, as the concentrations of the reactants has to be positive. Furthermore, for our purpose, it is enough to find a Lyapunov function that has non-positive derivative in a local region around the equilibrium. In this case, we can impose the inequality constraints $x_1^2 \leq \gamma u_3^2$, $x_2^2 \leq \gamma u_4^2$ where $0 < \gamma \leq 1$. Thus, our system (complete with the equality and inequality constraints) is described by

$$\begin{aligned}
\dot{x}_1 &= u_1 - (x_1 + u_3) + (x_1 + u_3)^2 (x_2 + u_4) \\
\dot{x}_2 &= u_2 - (x_1 + u_3)^2 (x_2 + u_4) \\
0 &\geq x_1^2 - \gamma u_3^2 \\
0 &\geq x_2^2 - \gamma u_4^2 \\
0 &\geq \underline{a} - u_1 \\
0 &\geq \underline{b} - u_2 \\
0 &= u_1 - u_3 + u_3^2 u_4 \\
0 &= u_2 - u_3^2 u_4.
\end{aligned}$$

For this example, two quartic Lyapunov functions, which prove stability of all the dynamical systems within the following ranges of a and b and which are parameterized by u_1 and u_2 have been constructed, using Proposition 9.

$$\text{Region 1:} \quad (\underline{a}, \underline{b}) = (0.25, 0) \tag{22}$$

$$\text{Region 2:} \quad (\underline{a}, \underline{b}) = (0, 1.05) \tag{23}$$

Each of them has more than 30 terms in it, and is therefore not listed here. However their level curves are shown for two different parameter values in Figure 3.

Dynamic uncertainty, on the other hand, can be characterized using integral constraints [26]. For example, an uncertain but $\mathcal{L}_2$-norm bounded feedback operator relating x and u can be represented by the IQC

$$\int_0^T (\gamma x^T x - u^T u) dt \geq 0.$$

Stability of the whole system can then be verified using Theorem 8.

2) Input-Output and Dissipativity Analysis: Input-output analysis within the framework of dissipative systems theory [56], [57] can also be addressed using the SOS technique. Let us consider the system

$$\dot{x} = f(x, u), \tag{24}$$

$$y = h(x, u), \tag{25}$$

where u and y are respectively the input and output of the system, and $f(x, u)$, $h(x, u)$ are polynomials in x and u. Dissipativity of the above system with respect to a polynomial supply rate function $w(u, y)$ can be verified using the following proposition.

Proposition 11: Suppose for the system (24)–(25) and a given supply rate function $w(u, y)$ there exists a SOS polynomial $S(x)$, such that $S(0) = 0$ and

$$w(u, h(x, u)) - \frac{\partial S}{\partial x} f(x, u) \text{ is SOS.} \tag{26}$$

Then the system is dissipative with respect to the supply rate function $w(u, y)$, as proven by the existence of a storage function $S(x)$.

Proof: The function $S(x)$ being an SOS implies that $S(x) \geq 0$. Now notice that Equation (26) implies

$$\frac{\partial S}{\partial x} f(x, u) \leq w(u, h(x, u)) \quad \forall u \text{ and } x,$$

which is the differential version of the dissipation inequality [56]. Since $S(x)$ satisfies the inequality, it follows that $S(x)$ is a storage function for the system, thus proving that the system is dissipative with respect to the supply rate $w(u, y)$. ■

An important choice of supply rate function is

$$w(u, y) = \gamma u^T u - y^T y,$$

since dissipativity of the system with respect to this supply rate implies that the $\mathcal{L}_2$-gain of the input-output map $u \mapsto y$ is less than or equal to $\sqrt{\gamma}$. Thus, by minimizing γ and solving the corresponding SOS problem, we can obtain an estimate of the $\mathcal{L}_2$-gain of the above nonlinear input-output map.

Example 12: Consider the system

$$\begin{aligned}
\dot{x}_1 &= -2x_1 - x_1^3 + x_1 x_2^2 + 5x_1 x_3 \\
\dot{x}_2 &= -3x_2 + x_1^2 - x_1 x_2 - x_1^2 x_2 - x_2^3 + x_1^2 x_3 + x_1 x_3^2 \\
\dot{x}_3 &= -2x_1^2 - x_1 x_2 - 0.5x_1^3 - 4x_3^3 + u \\
y &= x_1
\end{aligned}$$

We will compute an upper bound for the $\mathcal{L}_2$-gain of the map $u \mapsto y$ using the method describe in Subsection IV-C.2. A quadratic storage function for this system cannot be found, but a quartic one exists for $\gamma = 0.47$. This shows that the $\mathcal{L}_2$-gain of the system is less than or equal to $\sqrt{0.47} = 0.6856$.

V. Analysis of Hybrid Systems

Many systems have dynamics that are described by a set of continuous time differential equations in conjunction with a discrete event process. Such systems are usually referred to as switched or hybrid systems, and their stability analysis has been treated in [6], [18], [11].

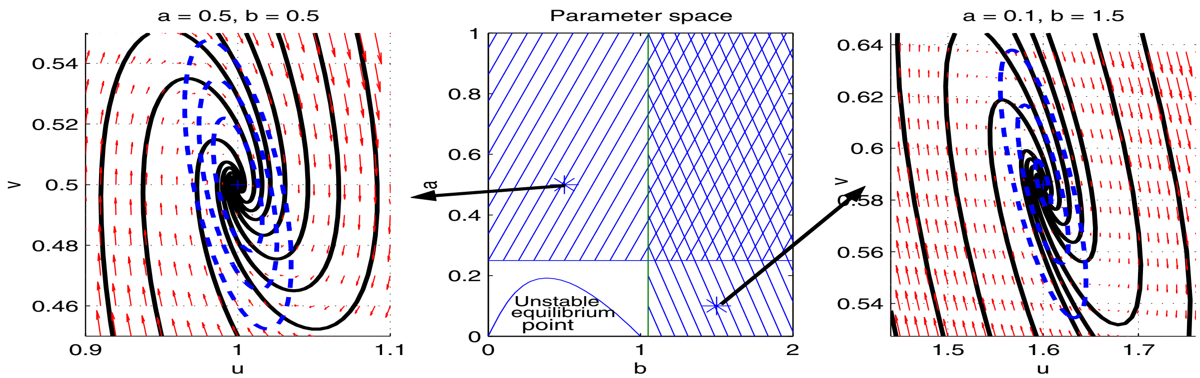

Fig. 3. Chemical Oscillator Example: Two parameter instances of the constructed parameterized Lyapunov functions. Arrows show the vector field, the level curves of Lyapunov functions are shown dashed, and a few trajectories are shown with solid lines.

One way of proving stability is by using piecewise quadratic Lyapunov functions [18], [16], which are constructed by concatenating several quadratic Lyapunov-like functions. This approach is quite effective, as the search for such Lyapunov functions can be performed by solving linear matrix inequalities (LMIs). However, in some cases it can be conservative.

In this section we will show how polynomial and piecewise polynomial Lyapunov functions can be constructed using the SOS decomposition. The method generalizes previous analysis methods using quadratic and piecewise quadratic Lyapunov functions. Some features of the new approach are:

- it provides a less conservative test for proving stability under arbitrary switching;
- stability can be proven with a smaller number of Lyapunov-like functions, eliminating the need of refining the state space partition;
- the method can be applied to systems with nonlinear subsystems and nonlinear switching surfaces;
- parametric robustness analysis can be performed in a straightforward manner.

A. Preliminaries and Notation

Here we establish the notation we will be using in the rest of this section. Consider systems of the form:

$$\dot{x} = f_i(x), \quad i \in I = \{1, ..., N\}, \tag{27}$$

where $x \in \mathbb{R}^n$ is the continuous state, i is the discrete state, $f_i(x)$ is the vector field describing the dynamics of the i-th mode/subsystem, and I is the index set. We assume that the origin is an equilibrium of the system.

Depending on how the discrete state i evolves, a system like (27) can be categorized as a switched system, if for each $x \in \mathbb{R}^n$ only one $i \in I$ is possible, or as a hybrid system, if for some $x \in \mathbb{R}^n$ multiple i are possible. The former type of systems includes systems with saturation and variable structure systems, whereas the latter type includes systems with hysteresis, systems with finite automata, etc.

More specifically, in the case of switched systems, the system is in the i-th mode at time t if $x(t) \in X_i$, where $X_i \subset \mathbb{R}^n$ is a region of the state space described by

$$X_i = \{x \in \mathbb{R}^n : g_{ik}(x) \geq 0, \text{ for } k = 1, ..., m_{X_i}\}, \tag{28}$$

for some $g_{ik} : \mathbb{R}^n \to \mathbb{R}$. Additionally, the state space partition $\{X_i\}$ must satisfy $\bigcup_{i \in I} X_i = \mathbb{R}^n$ and $\text{int}(X_i) \cap \text{int}(X_j) = \emptyset$ for $i \neq j$. A switching surface between the i-th and j-th modes, i.e. a boundary between X_i and X_j, is given by

$$S_{ij} = \{x : h_{ij0}(x) = 0, h_{ijk}(x) \geq 0, k = 1, ..., m_{S_{ij}}\}, \tag{29}$$

for some $h_{ijk} : \mathbb{R}^n \to \mathbb{R}$. Note that the transition between modes on this surface can occur in both directions. Although in principle the direction of transition for a particular $x \in S_{ij}$ can be determined from the vector fields $f_i(x)$ and $f_j(x)$, it is assumed in our analysis that such a characterization is not performed a priori.

On the other hand, the evolution of the discrete state in a hybrid system is governed by

$$i(t) = \phi(x(t), i(t^-)), \tag{30}$$

with $\phi : \mathbb{R}^n \times I \to I$. Corresponding to the transition law ϕ, there exists a region of the state space where a particular mode can be active. For the i-th mode, the active region is denoted by X_i, and is given by[4]

$$X_i = \{x \in \mathbb{R}^n : g_{ik}(x) \geq 0, \text{ for } k = 1, ..., m_{X_i}\}. \tag{31}$$

In the hybrid system case, $\bigcup_{i \in I} X_i = \mathbb{R}^n$ still holds,

[4]A notation similar to (28) is chosen here for simplicity; the interpretation should be clear from the context.

but $\text{int}(X_i) \cap \text{int}(X_j)$ is not necessarily empty for $i \neq j$. The transition set from the j-th mode to the i-th mode in a hybrid system is described by

$$\begin{aligned} S_{ij} &= \{x : i = \phi(x, j)\} \\ &= \{x : h_{ij0}(x) = 0, h_{ijk}(x) \geq 0, k = 1, ..., m_{S_{ij}}\}. \end{aligned} \tag{32}$$

In contrast to switched systems, the transition between modes on S_{ij} for a hybrid system occurs only in one direction, namely from j to i.

We assume that the discrete state $i(t)$ is piecewise continuous. Systems with infinitely fast switching, such as those that have sliding modes, are excluded from our discussion. We also assume that f_i, g_{ik}, and h_{ijk} are polynomials. In the case where any of these functions is nonpolynomial, we can use a recasting technique [33].

B. Stability Analysis

1) Stability Under Arbitrary Switching: We will first consider stability of the system (27) under arbitrary switching. A sufficient condition for such stability is the existence of a global common Lyapunov function for all f_i's, as summarized in the following theorem.

Theorem 13: Suppose that for the set of vector fields $\{f_i\}$ there exists a polynomial $V(x)$ such that $V(0) = 0$ and

$$V(x) > 0 \quad \forall x \neq 0, \tag{33}$$

$$\frac{\partial V}{\partial x} f_i(x) < 0 \quad \forall x \neq 0, \ i \in I, \tag{34}$$

then the origin of the state space of the system (27) is globally asymptotically stable under arbitrary switching. Notice in particular that if the vector fields are linear, i.e. $f_i(x) = A_i x$, and if $V(x)$ is chosen to be quadratic, say $V(x) = x^T P x$, then the conditions in Theorem 13 correspond to the well-known LMIs $P > 0$, $A_i^T P + PA_i < 0$ for all i, which prove quadratic stability of the system. For higher degree polynomial vector fields and Lyapunov functions, the search for $V(x)$ can also be performed using semidefinite programming by formulating the conditions as SOS conditions, as described in Section II. The higher degree test is generally less conservative than the quadratic test, as a higher degree Lyapunov functions may exist even if the system does not possess a quadratic Lyapunov function. At worst, these two tests have the same conservatism.

Example 14: Consider the system $\dot{x} = f_i(x)$, $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, with

$$f_1(x) = \begin{bmatrix} -5x_1 - 4x_2 \\ -x_1 - 2x_2 \end{bmatrix}, \quad f_2(x) = \begin{bmatrix} -2x_1 - 4x_2 \\ 20x_1 - 2x_2 \end{bmatrix}.$$

It can be proven using a dual semidefinite program that no global quadratic Lyapunov function exists for

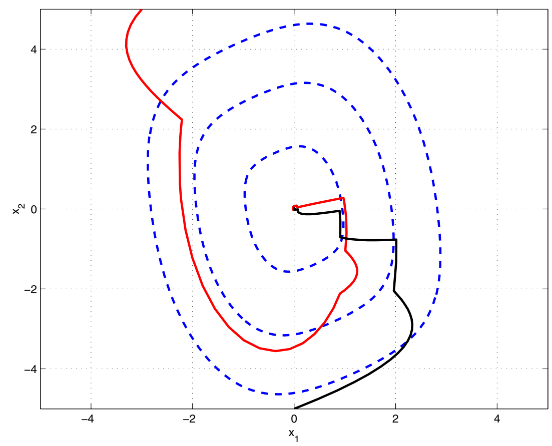

Fig. 4. Trajectories of the system in Example 14 under arbitrary switching. Dashed curves are level curves of the common Lyapunov function.

this system [18]. Nevertheless, a global sextic Lyapunov function

$$\begin{aligned} V(x) = 19.861x_1^6 &+ 11.709x_1^5 x_2 + 14.17x_1^4 x_2^2 \\ &+ 4.2277x_1^3 x_2^3 + 8.3495x_1^2 x_2^4 - 1.2117x_1 x_2^5 \\ &+ 1.0421x_2^6 \end{aligned}$$

exists, and therefore the system is asymptotically stable under arbitrary switching (cf. Figure 4).

2) Piecewise Polynomial Lyapunov Functions: Most switched and hybrid systems come with a prescribed switching scheme or a discrete transition rule. In this case, stability can be proven in a more effective way using piecewise polynomial Lyapunov functions. Such functions are concatenations of polynomial functions $V_i(x)$ (also termed Lyapunov-like functions), typically corresponding to the state space partition $\{X_i\}$. The Lyapunov-like function $V_i(x)$ and its time derivative along the trajectory of the i-th mode are required to be positive and negative respectively, only within X_i.

The conditions in the previous paragraph can be accommodated using a method similar to the S-procedure [5] as follows. To incorporate the fact that $V_i(x)$ only needs to be positive on X_i, where X_i is described by (28), we impose the relaxed condition

$$V_i(x) - \sum_{k=1}^{m_{X_i}} a_{ik}(x) g_{ik}(x) > 0, \tag{35}$$

for some $a_{ik}(x) \geq 0$. Since $g_{ik}(x)$ is nonnegative on X_i, the above condition implies that $V_i(x)$ is positive on X_i. An analogous condition can be imposed on $\frac{dV_i}{dt}$. Note that there is no requirement that the multipliers $a_{ik}(x)$ be constants (as in the S-procedure); they can also be polynomials of higher degree. Thus, our condition is generally less conservative than the S-procedure.

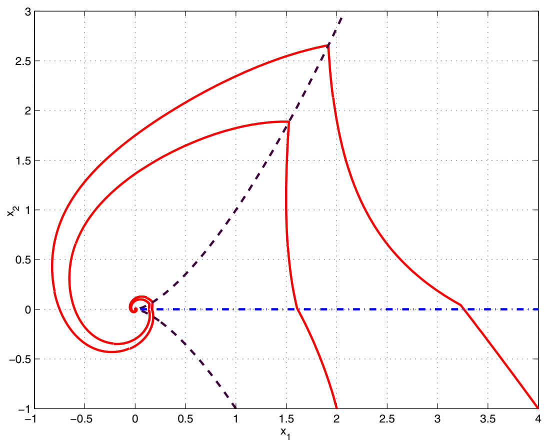

Fig. 5. Trajectories of the system in Example 15. Dash-dotted line and dashed curves show S_{21} and S_{12}, respectively.

3) Nonlinear Vector Fields and Switching Surfaces/Transition Sets: So far, the systems we have considered in the examples have linear subsystems and linear switching surfaces. As mentioned previously, the SOS conditions can be applied directly to systems with nonlinear vector fields and nonlinear switching surfaces or transition sets. To illustrate this, consider the following example.

Example 15: Let the hybrid system $\dot{x} = f_i(x)$ be composed of two subsystems

$$f_1(x) = \begin{bmatrix} -2x_1 - x_1^3 - 5x_2 - x_2^3 \\ 6x_1 + x_1^3 - 3x_2 - x_2^3 \end{bmatrix},$$
$$f_2(x) = \begin{bmatrix} x_2 + x_1^2 - x_1^3 \\ 4x_1 + 2x_2 \end{bmatrix},$$

with its transition rule given by

$$\phi(0) = 1,$$
$$\phi(t) = \begin{cases} 1, \text{ if } i(t-) = 2 \text{ and } x_2^2(t) = x_1^3(t), \\ 2, \text{ if } i(t-) = 1 \text{ and } x_2(t) = 0, \; x_1(t) \geq 0. \end{cases}$$

Figure 5 depicts some trajectories of the system. The active regions corresponding to the two modes are $X_1 = \mathbb{R}^2$ and $X_2 = \{x \in \mathbb{R}^2 : (x_1^3 - x_2^2) \geq 0\}$, while the transition sets are $S_{12} = \{x \in \mathbb{R}^2 : x_2^2 = x_1^3\}$ and $S_{21} = \{x \in \mathbb{R}^2 : x_2 = 0, x_1 \geq 0\}$. We can construct a sextic piecewise polynomial Lyapunov function given by

$$V(x(t)) = V_i(x(t)), \quad \text{if } \phi(t) = i,$$

for some $V_i(x)$'s, proving global asymptotic stability.

Even for a system with a rational or nonpolynomial vector field, a system embedding can sometimes be made such that a Lyapunov function that proves stability can be computed using the SOS decomposition. This has been presented in [32] and will not be discussed in this tutorial. The same technique can also be applied to nonpolynomial switching surfaces or transition sets.

Robust stability analysis of switched or hybrid systems can be treated using parameter dependent Lyapunov-like functions and multipliers. Computation of parameter dependent quadratic Lyapunov-like functions using LMIs had been previously difficult, since such functions are nonquadratic polynomials in the state and parameter variables. Using the SOS decomposition, computation of even higher degree functions is straightforward. A more detail description appeared in [42].

VI. Analysis of Time-Delay Systems

Significant progress has been made in the stability analysis of linear autonomous time-delay systems (TDS) using time-domain (*Lyapunov*) and frequency domain methods [15], [20]. In the linear case so called Lyapunov-Krasovskii (L-K) functionals are constructed by solving LMIs. On the other hand, the stability analysis of nonlinear time delay systems is far more difficult and so-called Lyapunov-Razumikhin (L-R) functions are usually constructed 'manually' in this case [22].

Here we present an extension of this methodology to the construction of L-K functionals for time-delay systems. The functionals that we use have structures that are similar to the complete functionals used for stability analysis of linear systems but they have kernels that are *polynomials*. This allows the use of the SOS decomposition to check the resulting stability conditions through the solution of LMIs. The methodology reduces to the standard LMI conditions when the system under consideration is linear and the functional has quadratic kernels. The same methodology can be used to analyze robust stability under parametric uncertainty.

The notation we will be using is standard, and is the one that is used in [15]. $\mathbb{R}^n$ is an n-dimensional real Euclidean space with norm $|\cdot|$. For $b > a$ denote $C([a,b], \mathbb{R}^n)$ the Banach space of continuous functions mapping the interval $[a,b]$ into $\mathbb{R}^n$ with the topology of uniform convergence. For $\phi \in C([a,b], \mathbb{R}^n)$ the norm of ϕ is defined as $\|\phi\| = \sup_{a \leq \theta \leq b} |\phi(\theta)|$, where $|\cdot|$ is a norm in $\mathbb{R}^n$. Also $C^\gamma = \{\phi \in C : \|\phi\| < \gamma\}$.

A. Stability Analysis of Time-delay Systems

We will be concerned with autonomous Retarded Functional Differential Equations (RFDEs) given by

$$\dot{x}(t) = f(x_t). \tag{36}$$

where $f : \Omega \to R^n$, $\Omega \subset C$, '$\dot{}$' represents the right-hand derivative and $x_t \in \Omega$, $x_t(\theta) = x(t+\theta)$, $\theta \in [-r, 0]$. Definitions of stability of the steady-state x^* of this system satisfying $f(x^*) = 0$ can be found in [15].

Assessing the stability properties of the equilibrium of (36) can be done using *time-domain* methodologies by constructing a Lyapunov-Krasovskii (L-K) functional. Let $\Omega \subset C^\gamma$, define $V : \Omega \to \mathbb{R}$ a continuous function and let $\dot{V}$ denote the *Upper Right Dini Derivative*. Then we have the following theorem [20]:

Theorem 16: (Lyapunov-Krasovskii) Let $\Omega \subset C^\gamma$. Suppose $V : \Omega \to \mathbb{R}$ is continuous and there exist nonnegative functions $a(s)$ and $b(s)$ such that $a(s) \to \infty$ as $s \to \infty$, and $a(0) = b(0) = 0$ such that

$$a(|\phi(0)|) \le V(\phi), \;\; \dot{V}(\phi) \le -b(|\phi(0)|) \; \forall \; \phi \in \Omega. \tag{37}$$

Then the solution $x = 0$ of (36) is uniformly stable. If, in addition, $b(s)$ is positive definite, then the solution $x = 0$ of (36) is uniformly asymptotically stable.

Just as in the case of ODEs, this is a powerful theorem as it answers questions about stability without requiring a solution to (36). At the same time, however, no methodology exists to construct these functions. Here we will use the SOS decomposition and construct Lyapunov functionals V with *polynomial kernels*, sacrificing non-negativity of V with the non-negativity of its kernel. Non-polynomial FDEs can be handled in a way similar to non-polynomial ODEs [32].

Consider the following functional:

$$\begin{aligned} V(x_t) \;\; = \;\; & V_0(x(t)) + \int_{-r}^{0} V_1(\theta, x(t), x(t+\theta))d\theta + \\ & + \int_{-r}^{0}\int_{t+\theta}^{t} V_2(x(\zeta))d\zeta d\theta \end{aligned} \tag{38}$$

for the system of the form (36). The first term is added to impose positive definiteness of V and the last term is added for convenience, as it will be used in the derivative condition to 'complete the squares'. Sufficient conditions for the (global) stability of the zero equilibrium can then be formulated as follows:

Proposition 17: Let 0 be an equilibrium for the system given by (36). Let there exist polynomials V_0, V_1 and V_2 and a positive definite polynomial $\varphi(x(t))$ such that:

1) $V_0(x(t)) - \varphi(x(t)) \ge 0$,
2) $V_1(\theta, x(t), x(t+\theta)) \ge 0$ for $\theta \in [-r, 0]$,
3) $V_2(x(\zeta)) \ge 0$,
4) $r\frac{\partial V_1}{\partial x(t)} f + \frac{dV_0}{dx(t)} f - r\frac{\partial V_1}{\partial \theta} + rV_2(x(t)) - rV_2(x(t+\theta)) + V_1(0, x(t), x(t)) - V_1(-r, x(t), x(t-r)) \le 0$ for $\theta \in [-r, 0]$.

Then the equilibrium 0 of the system given by (36) is *globally uniformly stable*.
A proof can be found in [30].

This proposition can be used in practice in a similar way as described in the delay-independent case. To impose the conditions $\theta \in [-r, 0]$, we use a process similar to the S-procedure, as it was done in Theorem 8. The polynomial $V_1(\theta, x(t), x(t+\theta))$ is required to be non-negative only when $h(\theta) = \theta(\theta + r) \le 0$ is satisfied. We therefore adjoin this constraint to a, using instead of constant positive multipliers (S-procedure), SOS multipliers p, and we rewrite condition (2) in Proposition 17 above, as follows:

$V_1(\theta, x(t), x(t+\theta)) + p(\theta, x(t), x(t+\theta))h(\theta)$ is a SOS

Condition (4) can be verified in a similar manner. This results in four SOS conditions in a relevant SOS programme which can be solved using SOSTOOLS [44]. We can also consider different Lyapunov structures [30].

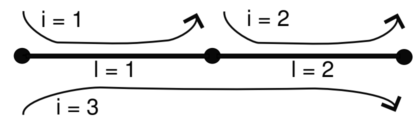

Fig. 6. A simple network.

Remark 18: As remarked earlier, when dealing with nonlinear systems with multiple equilibria or with natural constraints on their state-space, it is useful to use a restricted region for which stability is to be proven, in the same way that it was done in the delay-independent case. We will still need to specify $\Omega = \{x_t \in C : \|x_t\| \le \gamma\}$, and adjoin the relevant conditions on $x(t)$, $x(t-r)$ and $x(t+\theta) \; \forall \; \theta \in [-r, 0]$ to the relevant kernels of the Lyapunov functionals using the extended S-procedure, in much the same way that the conditions $\theta \in [-r, 0]$ were adjoined in Conditions (2) and (4) of Proposition 17.

B. Example: Stability Analysis of A Network Congestion Control Scheme

Consider the network shown in Figure 6 that uses a primal-dual version of FAST [25] as its protocol. We assume that all forward and backward delays are overbounded by $\tau/2$, and that the sources and links have dynamics that are described in [29]. The system is not polynomial in its original form, but it can be rendered polynomial through some nonlinear transformations [31]. The closed loop system is then given by

$$\dot{z}_1(t) = \begin{pmatrix} -\frac{K_1\beta}{\tilde{K}_1 c\tau}[z_1(t) + z_4(t) + z_1(t)z_4(t)] \\ -\frac{\alpha}{\tau}[z_1(t)+1][\tilde{K}_2 z_3(t-\tau) + \tilde{K}_1 z_1(t-\tau)] \end{pmatrix}$$
$$\dot{z}_2(t) = \begin{pmatrix} -\frac{K_2\beta}{\tilde{K}_1 c\tau}[z_2(t) + z_5(t) + z_2(t)z_5(t)]+ \\ -\frac{\alpha}{\tau}[z_2(t)+1][\tilde{K}_2 z_3(t-\tau) + \tilde{K}_1 z_2(t-\tau)] \end{pmatrix}$$

$$\dot{z}_3(t) = \begin{pmatrix} -\frac{\beta}{\tilde{K}_1 c\tau}[(z_3z_4 + z_3 + z_4)K_1 + (z_3z_5 + z_3 + z_5)K_2] \\ -\frac{\alpha}{2\tau}[z_3(t) + 1] \\ \times[z_1(t-\tau)\tilde{K}_1 + z_2(t-\tau)\tilde{K}_1 + 2z_3(t-\tau)\tilde{K}_2] \end{pmatrix}$$

$$\dot{z}_4(t) = \frac{\tilde{K}_1 c}{K_1}(\tilde{K}_1 z_1(t-\tau) + \tilde{K}_2 z_3(t-\tau))$$

$$\dot{z}_5(t) = \frac{\tilde{K}_1 c}{K_2}((K_1 + K_2)z_2(t-\tau) + K_3 z_3(t-\tau))$$

We set $c = 40$, $\alpha = 1$, $\tau = 0.2$, and we calculate $\beta = \frac{0.64\alpha}{\tau}$ and we let $K_1 = 15$, $K_2 = 20$, $K_3 = 25$; furthermore, $\tilde{K}_1 = \frac{K_1+K_2}{K_1+K_2+K_3}$ and $\tilde{K}_2 = \frac{K_3}{K_1+K_2+K_3}$. We can construct a similar Lyapunov functional to (38) with all polynomials V_0, V_1 of second order and V_2 of order 4 for

$$0 \le x_{1_t} \le 2.3x_{1,0}, \quad 0 \le x_{2_t} \le 2.3x_{2,0},$$
$$0 \le x_{3_t} \le 2.3x_{3,0}, \quad q_1 \ge 0, \quad q_2 \ge 0.$$

This proves stability of the equilibrium for $\tau = 0.2$.

VII. Model Validation

Modelling is an important precursor to system analysis and controller design. For successful analysis and design, it is crucial to obtain a model that captures essential behaviors of the system under consideration. Model validation provides a way to evaluate the ability of a proposed model to represent observed system behaviors. However, as often mentioned in the literature [51], [37], [12], "model validation" is actually a misnomer; it is impossible to validate a model, because to do so requires an infinite number of experiments and data. The role of model validation is to *invalidate* a model, by proving that some experimental data are inconsistent with the model, thus indicating that a refinement of the model is required.

In the simplest setting, consider the model

$$\dot{x}(t) = f(x(t), p, t), \tag{39}$$

where $x(t) \in \mathbb{R}^n$ is the vector of state variables, t is the time, and $p \in \mathbb{R}^m$ is the parameter vector, assumed to take its value in a set $P \subset \mathbb{R}^m$. Let an experiment be performed with the real system, and two measurements be taken at time $t = 0$ and $t = T$. Suppose that these measurements indicate that $x(0) \in X_0$ and $x(T) \in X_T$, where both X_0 and X_T are subsets of $\mathbb{R}^n$. In addition, assume that $x(t) \in X$ for all $t \in [0, T]$, where $X \subseteq \mathbb{R}^n$. With these notations, the invalidation problem can be stated as follows:

Problem 19: Given the model (39), parameter set P, and trajectory information $\{X_0, X_T, X\}$, provide a proof that the model (39) with parameter set P is inconsistent with $\{X_0, X_T, X\}$. That is, prove that for all possible parameter $p \in P$, the model (39) cannot produce a trajectory $x(t)$ such that $x(0) \in X_0$, $x(T) \in X_T$, and $x(t) \in X, \forall t \in [0, T]$.

Before proceeding further, we would like to remark that necessarily $X_0 \subseteq X$ and $X_T \subseteq X$, and in most cases X will be *much larger* than X_0 or X_T. In fact, X can be the whole state space. The information about X may come from the experiment and/or from *a priori* knowledge about the system[5], and such information will strengthen the model validation test. Note also that output measurement using the output $y = g(x)$ can be accommodated, e.g. by defining $X_0 = \{x \in X : \underline{y}_0 \le g(x) \le \overline{y}_0\}$, and similarly for X_T.

If such a proof in Problem 19 can be found, then we say that the model (39) and parameter set P are invalidated by $\{X_0, X_T, X\}$. Traditional approaches for solving this problem include exhaustive simulation of (39) using parameters p and initial conditions $x(0)$ sampled randomly from P and X_0. If after many such simulations no trajectory $x(t)$ that satisfies the initial hypothesis can be found, then inconsistency is concluded. Indeed simulation (possibly after some parameter fitting) is a good way for proving that a model can reproduce *some* behaviors of the system it represents. However, for proving inconsistency, the required number of simulation runs will soon become prohibitive. Moreover, a proof by simulation alone is *never exact*, simply because it is impossible to test all p and $x(0)$.

On the other hand, our method relies on the existence of a function of state-parameter-time, which we term barrier certificate. A barrier certificate gives an *exact* proof of inconsistency by providing a barrier between possible trajectories of the model starting at X_0 and the final measurement X_T. This is accomplished without performing any simulation nor computing the flow of the model. The method is summarized in the following theorem.

Theorem 20 ([39]): Let the model (39) and the sets P, X_0, X_T, X be given. Assume that there is a function $B : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ (a barrier certificate), differentiable with respect to x and t, such that

$$B(x_T, p, T) - B(x_0, p, 0) > 0 \qquad \forall x_T \in X_T, x_0 \in X_0, p \in P, \tag{40}$$

$$\frac{\partial B}{\partial x}(x, p, t) f(x, p, t) + \frac{\partial B}{\partial t}(x, p, t) \le 0 \qquad \forall x \in X, p \in P, t \in [0, T]. \tag{41}$$

Then the model (39) and its associated parameter set P are invalidated by $\{X_0, X_T, T\}$.

Example 21: As the second example, consider the model $\dot{x} = -px^3$, with $X = \mathbb{R}$ and $P = [0.5, 2]$. The measurement data used for invalidating this model are $X_0 = [0.85, 0.95]$ and $X_T = [0.55, 0.65]$ at $T = 4$. Us-

[5]For example, in biological systems typical state variables are the concentration of some chemical substrates. In this case, they can be neither negative nor very large.

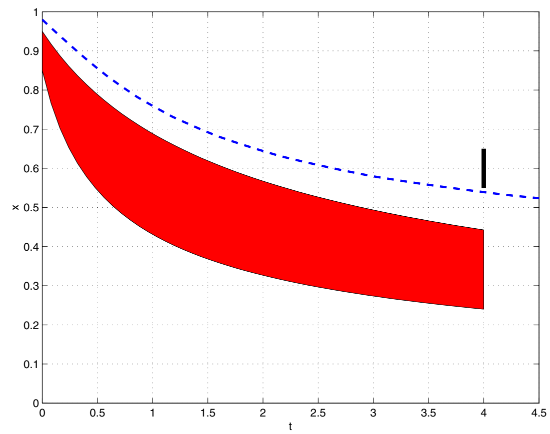

Fig. 7. A level set of the barrier certificate $B(x,t)$ in Example 21 is shown as a dashed curve in this figure. Bold line at $t=4$ is X_T, whereas the solid patch is the collection of all possible trajectories of the model with $p \in P$, starting at $x(0) \in X_0$.

ing the SOS technique, we obtain as a barrier certificate for this example $B(x,t) = 8.35x + 10.4x^2 - 21.5x^3 + 9.86x^4 - 1.78t + 6.58tx - 4.12tx^2 - 1.19tx^3 + 1.54tx^4$. It is shown in Figure 7 how a level set of this function serves as a barrier in the state-time space.

With this methodology, we are able to treat in a unified way model validation of a very large class of continuous-time models — some of which have never been addressed before. This includes differential-algebraic models, models with uncertain inputs, models with memoryless and dynamic uncertainties, hybrid models, and their combinations. Moreover, the methods are computationally tractable, as barrier certificates can be constructed using the SOS decomposition, in a way similar to what we have shown in the previous sections. These we consider as some of the most important features of our approach. For more on these, including how to extend the invalidation setting to the case where there are measurements at more than two time instants, see [39].

VIII. Safety Verification

Complex behaviors that can be exhibited by modern engineering systems, which typically have hybrid (i.e., a mixture of discrete and continuous) dynamics, make the safety verification of such systems both critical and challenging. In principle, safety verification or reachability analysis aims to show that starting at some initial conditions, a system cannot evolve to some unsafe region in the state space. For safety verification, several methods have been proposed (see e.g. [3], [2], [8], [54]). Explicit computation of either exact or approximate reachable sets corresponding to the continuous dynamics is crucial for virtually all of these methods. Consequently, it is hard to handle nonlinearity, uncertainty, and constraints with these methods.

With regard to this, it is interesting to note that safety verification addresses a question related to Problem 19 in the previous section. If we assume that the set X_T in the previous section is the unsafe set, then verifying safety requires proving that no trajectory of the system starting from X_0 enters this set for all positive time instant. Thus, it is natural to expect that barrier certificates can also be used for safety verification, and in fact it is.

The safety verification method based on barrier certificates can be easily adapted to handle hybrid systems, as we will show in this section. We adopt the hybrid modelling framework that was first proposed in [1], which is more general than those in Section V. See also [2] for a more detailed explanation and example. A hybrid system is a tuple $H = (\mathcal{X}, L, X_0, I, F, T)$ with the following components:

- $\mathcal{X} \subseteq \mathbb{R}^n$ is the continuous state space.
- L is a finite set of locations. The overall state space of the system is $X = L \times \mathcal{X}$, and a state of the system is denoted by $(l,x) \in L \times \mathcal{X}$.
- $X_0 \subseteq X$ is the set of initial states.
- $I : L \to 2^{\mathcal{X}}$ is the invariant, which assigns to each location l an invariant set $I(l) \subseteq \mathcal{X}$ that contains all possible continuous states while at location l.
- $F : X \to 2^{\mathbb{R}^n}$ is a set of vector fields. F assigns to each $(l,x) \in X$ a set $F(l,x) \subseteq \mathbb{R}^n$ which constrains the evolution of the continuous state according to the differential inclusion $\dot{x} \in F(l,x)$.
- $T \subseteq X \times X$ is a relation capturing discrete transitions between two locations. Here a transition $((l,x),(l',x')) \in T$ indicates that from the state (l,x) the system can undergo a discrete jump to the state (l',x').

Trajectories of the hybrid system H start from some initial state $(l_0,x_0) \in X_0$ and are concatenations of a sequence of continuous flows and discrete transitions. During a continuous flow, the discrete location l is maintained and the continuous state evolves according to the differential inclusion $\dot{x} \in F(l,x)$, as long as x remains inside the invariant set $I(l)$. At a state (l_1,x_1), a discrete transition to (l_2,x_2) can occur if $((l_1,x_1),(l_2,x_2)) \in T$. Given a hybrid system H and a set of unsafe states $X_u \subseteq X$, the safety verification problem is concerned with proving that all trajectories of the hybrid system H cannot enter the unsafe region X_u.

For each location $l \in L$, we define the set of initial and unsafe continuous states as $\text{Init}(l) = \{x \in \mathcal{X} : (l,x) \in X_0\}$ and $\text{Unsafe}(l) = \{x \in \mathcal{X} : (l,x) \in X_u\}$. To each tuple $(l,l') \in L \times L$ with $l \neq l'$, we associate a guard set $\text{Guard}(l,l') = \{x \in \mathcal{X} : ((l,x),(l',x')) \in T \text{ for some } x' \in \mathcal{X}\}$, and a (possibly set valued) reset map $\text{Reset}(l,l') : x \mapsto \{x' \in \mathcal{X} : ((l,x),(l',x')) \in T\}$, whose domain is $\text{Guard}(l,l')$. Obviously, if no discrete transition from location l to location l' is possible, then the set $\text{Guard}(l,l')$ will be regarded as empty, and the associated reset map needs not be defined.

Using this formalism, the following test for safety can be stated.

Theorem 22 ([40]): Let the hybrid system $H = (\mathcal{X}, L, X_0, I, F, T)$ and the unsafe set X_u be given. Suppose there exists a collection of differentiable functions $B_l(x)$ which, for each $l \in L$ and $(l,l') \in L^2$, $l' \neq l$, satisfy

$$B_l(x) > 0 \quad \forall x \in \textit{Unsafe}(l), \tag{42}$$

$$B_l(x) \leq 0 \quad \forall x \in \textit{Init}(l), \tag{43}$$

$$\frac{\partial B_l}{\partial x}(x) f_l(x,d) \leq 0 \quad \forall (x,d) \in I(l) \times D(l), \tag{44}$$

$$B_{l'}(x') \leq 0 \quad \forall x' \in \textit{Reset}(l,l')(x), \quad \text{for all } x \in \textit{Guard}(l,l') \text{ s.t. } B_l(x) \leq 0. \tag{45}$$

Then the safety of the hybrid system H is guaranteed.

Again, when the vector fields of the system are polynomials and the sets in the system description are semialgebraic (i.e., described by polynomial equalities and inequalities), the SOS technique can be utilized for constructing a polynomial barrier certificate $\{B_l(x)\}$. While the computational cost of this construction depends on the degrees of the vector fields and the barrier certificate in addition to the dimension of the continuous state, for fixed degrees the complexity is polynomial with respect to the state dimension. Hence we expect our method to be more scalable than many other existing safety verification methods.

A large class of hybrid systems can be treated within this framework, including those with nonlinear continuous dynamics, uncertain inputs, uncertain parameters, and constraints. More recently, the method has also been extended to handle stochastic safety verification [41]. For an application example, we refer the reader to [14].

IX. Conclusions

In this paper we have presented a brief tutorial on sum of squares techniques for systems analysis. We have shown how it can be used to solve problems such as nonlinear stability and robustness analysis, analysis of hybrid systems, analysis of time-delay systems, model validation, and safety verification. Other work in this area includes estimation of the domain of attraction [34], [50], LPV analysis and synthesis [58] and nonlinear synthesis [17], [47], [46]. We would also refer the reader to the paper [35] and the upcoming volume [13]. For industrial application examples, we refer the reader to [14], [21], [23], [9].

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Oliviero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[2] R. Alur, T. Dang, and F. Ivancic. Progress on reachability analysis of hybrid systems using predicate abstraction. In *Hybrid Systems: Computation and Control, LNCS 2623*, pages 4–19. Springer-Verlag, 2003.

[3] A. Bemporad, F. D. Torrisi, and M. Morari. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In *Hybrid Systems: Computation and Control, LNCS 1790*, pages 45–58. Springer-Verlag, 2000.

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

[5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (SIAM), 1994.

[6] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[7] M. D. Choi, T. Y. Lam, and B. Reznick. Sum of squares of real polynomials. *Proceedings of Symposia in Pure Mathematics*, 58(2):103–126, 1995.

[8] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Trans. Automatic Control*, 48(1):64–75, 2003.

[9] R. Cogill and S. Lall. Decentralized stochastic decision problems and polynomial optimization. In *Proceedings of the American Control Conference*, 2005.

[10] L. Dai. *Singular Control Systems*. Springer-Verlag, New York, 1989.

[11] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7):1069–1082, 2000.

[12] G. Dullerud and R. Smith. A nonlinear functional approach to LFT model validation. *Systems and Control Letters*, 47(1):1–11, 2002.

[13] A. Garulli and D. Henrion, editors. *Positive Polynomials in Control*. Springer–Verlag, 2005. Forthcoming.

[14] S. Glavaski, A. Papachristodoulou, and K. Ariyur. Controlled hybrid system safety verification: Advanced life support system testbed. In *Proceedings of the American Control Conference*, 2005.

[15] J. K. Hale and S. M. V. Lunel. *Introduction to Functional Differential Equations*. Applied Mathematical Sciences (99). Springer-Verlag, 1993.

[16] A. Hassibi and S. P. Boyd. Quadratic stabilization and control of piecewise-linear systems. In *Proceedings of American Control Conference*, pages 3659–64, 1998.

[17] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard. Some controls applications of sum of squares programming. In *Proceedings of the IEEE Conference on Decision and Control*, 2003.

[18] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):555–559, 1998.

[19] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Inc., second edition, 1996.

[20] V. Kolmanovskii and A. Myshkis. *Introduction to the Theory and Applications of Functional Differential Equations*. Kluwer Academic Publishers, 1999.

[21] K. Krishnaswamy, G. Papageorgiou, S. Glavaski, and A. Papachristodoulou. Analysis of aircraft pitch axis stability augmentation system using sum of squares optimization. In *Proceedings of the American Control Conference*, 2005.

[22] Y. Kuang. *Delay Differential Equations with Applications in Population Dynamics*. Academic Press, 1993. Mathematics in Science and Engineering (191).

[23] A. Lakshmikantha, C. L. Beck, and R. Srikant. On the use of SoS methods for analysis of connection-level stability in the Internet. In *Proceedings of the American Control Conference*, 2005.

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

[25] S. Low. Fast project. `http://netlab.caltech.edu/FAST/`, 2004.

[26] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

[27] J. D. Murray. *Mathematical Biology*. Springer-Verlag, second edition, 1993.

[28] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[29] F. Paganini, Z. Wang, S. H. Low, and J. C. Doyle. A new TCP/AQM for stable operation in fast networks. In *Proceedings of IEEE Infocom, San Francisco*, 2003.

[30] A. Papachristodoulou. Analysis of nonlinear time delay systems using the sum of squares decomposition. In *Proceedings of the American Control Conference*, 2004.

[31] A. Papachristodoulou, J. C. Doyle, and S. H. Low. Analysis of nonlinear delay differential equation models of TCP/AQM protocols using sums of squares. In *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[32] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of the IEEE Conference on Decision and Control*, 2002.

[33] A. Papachristodoulou and S. Prajna. Analysis of non-polynomial systems using the sum of squares decomposition. To appear in *Positive Polynomials in Control*, Springer-Verlag, 2005.

[34] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, Pasadena, CA, 2000.

[35] P. A. Parrilo and S. Lall. Semidefinite programming relaxations and algebraic optimization in control. *European Journal of Control*, 9(2–3):307–321, 2003.

[36] P. A. Parrilo and B. Sturmfels. Minimizing polynomial functions. In *Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science*, 2001.

[37] K. Poolla, P. Khargonekar, A. Tikku, J. Krause, and K. Nagpal. A time-domain approach to model validation. *IEEE Trans. Auto. Control*, 39(5):951–959, 1994.

[38] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Linear Algebra*, 127:99–104, 1998.

[39] S. Prajna. Barrier certificates for nonlinear model validation. In *Proceedings of the IEEE Conference on Decision and Control*, 2003.

[40] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, pages 477 – 492. Springer-Verlag, 2004.

[41] S. Prajna, A. Jadbabaie, and G. J. Pappas. Stochastic safety verification using barrier certificates. In *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[42] S. Prajna and A. Papachristodoulou. Analysis of switched and hybrid systems - beyond piecewise quadratic methods. In *Proceedings of the American Control Conference*, pages 2779–2784, 2003.

[43] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Proceedings of the IEEE Conference on Decision and Control*, 2002.

[44] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. SOSTOOLS – Sum of Squares Optimization Toolbox, User's Guide. Available at `http://www.cds.caltech.edu/sostools`, 2002.

[45] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. New developments in sum of squares optimization and SOSTOOLS. In *Proceedings of the American Control Conference*, 2004.

[46] S. Prajna, A. Papachristodoulou, and F. Wu. Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach. In *Proceedings of the Asian Control Conference*, 2004.

[47] S. Prajna, P. A. Parrilo, and A. Rantzer. Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, 49(2):310–314, 2004.

[48] A. Prestel and C. N. Delzell. *Positive Polynomials: From Hilbert's 17th Problem to Real Algebra*. Springer-Verlag, New York, 2001.

[49] B. Reznick. Some concrete aspects of Hilbert's 17th problem. In *Contemporary Mathematics*, volume 253, pages 251–272. American Mathematical Society, 2000.

[50] P. Seiler. Stability region estimates for SDRE controlled systems using sum of squares optimization. In *Proceedings of the American Control Conference*, 2003.

[51] R. S. Smith and J. C. Doyle. Model validation: a connection between robust control and identification. *IEEE Trans. Automatic Control*, 37(7):942–952, 1992.

[52] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Available at `http://fewcal.kub.nl/sturm/software/sedumi.html`.

[53] K. C. Toh, R. H. Tütüncü, and M. J. Todd. SDPT3 — a MATLAB software package for semidefinite-quadratic-linear programming, 1999. Available at `http://www.math.nus.edu.sg/ mattohkc/sdpt3.html`.

[54] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proc. of the IEEE*, 91(7):986–1001, 2003.

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

[56] J. C. Willems. Dissipative dynamical systems. I. General theory. *Arch. Rational Mech. Anal.*, 45:321–351, 1972.

[57] J. C. Willems. Dissipative dynamical systems. II. Linear systems with quadratic supply rates. *Arch. Rational Mech. Anal.*, 45:352–393, 1972.

[58] F. Wu and S. Prajna. A new solution approach to polynomial LPV system analysis and synthesis. In *Proceedings of the American Control Conference*, 2004.

[59] V. A. Yakubovic. S-procedure in nonlinear control theory. *Vestnik Leningrad University*, 4(1):73–93, 1977. English translation; original Russian publication in *Vestnik Leningradskogo Universiteta, Seriya Matematika*, Leningrad, Russia, 1971, pp. 62–77.