Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Relaxing Global Decrescence Conditions in Lyapunov Theorems for Hybrid Systems

Victor Shia * Robert Peter Matthew * Ram Vasudevan **
Ruzena Bajcsy *

* *Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA*
{vshia, rpmatthew, bajcsy} @eecs.berkeley.edu
** *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
ramv@mit.edu

Abstract: Lyapunov methods have been successfully applied to compute the Region of Attraction (ROA) of a system, i.e. the set of states that remain invariant for all time. A popular method for constructing Lyapunov functions is the Sum of Squares (SOS) approach. Current methods for determining stability of hybrid systems using SOS require either stability over the entire hybrid guard or knowledge of the switching times, limiting the types of systems that can be analyzed. In this paper, we introduce a new method to relax these constraints by only requiring decrescence properties over the 'active' subsections of each guard. We introduce our new formulation, explore useful modeling techniques, then apply them to five examples.

1. INTRODUCTION

Stability analysis and verification of hybrid systems often involve finding Lyapunov functions as certificates of stability or safety. For hybrid systems, stability must be defined simultaneously over all modes rather than treating each mode as a distinct dynamical system [Branicky, 1994].

One of the first results of applying Lyapunov theory to hybrid systems was to find a common Lyapunov function for a hybrid system [Liberzon and Morse, 1999]. Multiple Lyapunov functions and piecewise Lyapunov functions were also introduced as tools to determine the stability of switched and hybrid systems [Branicky, 1998, Johansson and Rantzer, 1998]. A survey of Lyapunov or Lyapunov-like functions applied to hybrid systems can be found in [Shorten et al., 2007].

Lyapunov conditions involving positive of functions are generally NP-hard to test [Murty and Kabadi, 1987]. However, by looking over a polynomial vector fields and Lyapunov functions, a function being Sum of Squares (SOS) is a checkable condition for positivity. This can be tested using semidefinite programming for which many solvers exist [Sturm, 1999, Lofberg, 2004, MOSEK, 2002, Parrilo, 2000]. For classical dynamical systems, SOS programming has been widely applied to the synthesis of Lyapunov functions for local stability analysis, region of attraction computation and robustness analysis [Papachristodoulou and Prajna, 2002, Jarvis-Wloszek et al., 2005, Papachristodoulou and Prajna, 2005, Tan, 2008].

Current approaches for hybrid system use SOS to find Lyapunov functions for stability [Papachristodoulou and Prajna, 2005, 2009, 2002], polynomial barrier certificates [Prajna and Rantzer, 2007], stability of limit cycles [Manchester et al., 2010], and control synthesis [Majumdar et al., 2012]. [Posa et al., 2013] computed ROAs for hybrid systems using the complimentary modeling framework. However, these methods require either a common equilbra across all of the modes, explicit knowledge of the switching times of the system, or global decrescence over an entire guard, restricting the applicability of these methods.

In this paper, we present a local Lyapunov theorem that removes the necessity of a single common equilibria, knowledge of the switching times and global decresence over the entire guard. We describe a computational approach using SOS programming to find local Lyapunov functions to determine an inner-approximation of the ROA of hybrid systems and apply this SOS program on several examples.

This paper is organized as follows. In Section 2, we present some background on hybrid systems, stability and SOS. In Section 3, we present a formulation for the local stability analysis of hybrid systems. In Section 4, we present several of the modeling techniques used. In Section 5, we present several examples and conclude in Section 6.

2. PRELIMINARIES

In this section, we introduce notation used throughout the remainder of the paper. We also include a brief introduction to hybrid systems, Lyapunov theory, SOS decomposition, and the S-procedure. Throughout this paper, let $x \in D$, where D is the space, usually $\mathbb{R}^n$. Let $\mathcal{R}[x]$ be the set of all polynomials in x and $\Sigma[x]$ be the set of all SOS polynomials in x and let $\mathbb{R}_+$ be the set of non-negative real numbers.

2.1 Hybrid Systems

A hybrid system is a dynamical system which combines both continuous and discrete dynamics. We first define the type of hybrid system we consider in this paper:

Copyright © 2014 IFAC

Definition 1. A hybrid dynamical system is a tuple $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$ where:

- $\mathcal{Q}$ is a finite set of discrete states of $\mathcal{H}$.
- $\mathcal{E} \subset \mathcal{Q} \times \mathcal{Q}$ is a set of edges forming a directed graph structure over $\mathcal{Q}$.
- $\mathcal{D} = \{D_q\}_{q \in \mathcal{Q}}$ is a set of domains and for each q, D_q is a subset of $\mathbb{R}^{n_q}$, $n_q \in \mathbb{N}$.
- $\mathcal{F} = \{f_q\}_{q \in \mathcal{Q}}$ is a set of vector fields, such that each $f_q : \mathbb{R} \times D_q \to D_q$ is a vector field defined on D_q.
- $\mathcal{G} = \{G_{(q,q')}\}_{(q,q') \in \mathcal{E}}$ is a set of guards where each $\{x | G_{(q,q')}(x) = 0\}$ defines a switching surface going from discrete state q to q'.
- $\mathcal{R} = \{R_{(q,q')}\}_{e \in \mathcal{E}}$ is a set of reset maps where each $R_{(q,q')} : D_q \to D_{q'}$ defines a state transition from q to q'.

Definition 2. $(q_e, x_e) \in \mathbb{R}^n$ is an equilibrium point for the system $\mathcal{H}$ if $f_{q_e}(t, x_e) = 0$ for all t.

Definition 3. $x_e \in \mathbb{R}^n$ is a common equilibrium point for the system $\mathcal{H}$ if for all $q \in \mathcal{Q}$, $x_e \in D_q$ and $f_q(t, x_e) = 0$ for all t.

In this paper, we make the following assumptions on hybrid systems.

Assumption 1. Each vector field f_q is piecewise continuous in its first argument and Lipschitz continuous in it second argument.

Assumption 2. Each vector field f_q, guard $G_{(q,q')}$ and reset map $R_{(q,q')}$ are polynomials.

Assumption 1 ensures existence and uniqueness of the solution of the differential equation for each discrete state. Assumptions 2 ensures conditions in Equation (12) are polynomials.

Assumption 3. One of the domains D_q contains the origin and the origin is an equilibrium point.

If the equilibrium is not at the origin, a state change can be performed to make the origin an equilibrium point. This assumption is made to simplify presentation.

2.2 Lyapunov Techniques for Hybrid Systems

We assume the reader is familiar with Lyapunov theory for continuous systems. An introduction can be found in [Khalil, 2002, Sastry, 1999].

Definition 4. U is said to be *invariant* if all trajectories that start in U stay in U for all forward time.

Existing theorems for Lyapunov theory for hybrid systems have been studied in [Branicky, 1998, Cai et al., 2008, Goebel and Teel, 2010, Papachristodoulou and Prajna, 2009, Shorten et al., 2007]. A classical theorem for Lyapunov stability of hybrid systems involves finding a "common" Lyapunov function across all the discrete states:

Theorem 1. Consider a hybrid system $\mathcal{H}$ defined as in Definition 1 where each D_q is a subset of $\mathbb{R}^n$. Let x_e be a common equilibrium point and let $R_{(q,q')}$ be the identity map. If there exists a C^1 function $V : \mathcal{D} \to \mathbb{R}$ such that:

(1) $V(x_e) = 0$ and $V(x) > 0, \forall x \in \mathcal{D} \setminus \{x_e\}$
(2) $\frac{\partial V}{\partial x} f_q(x) \leq 0$, $\forall x \in D, q \in \mathcal{Q}$

then x_e is a stable equilibrium point of $\mathcal{H}$. Moreover, if $\frac{\partial V}{\partial x} f_q(x) < 0 \quad \forall x \in D, q \in \mathcal{Q}$, then x_e is an asymptotically stable equilibrium point of $\mathcal{H}$.

The proof can be found in [Liberzon and Daniel, 2003]. Finding such a V would show stability for arbitrary switching but is difficult to find. To address this issue, Branicky introduced the idea of finding multiple Lyapunov functions for a hybrid system [Branicky, 1998]. The multiple Lyapunov function technique requires explicit knowledge of the switching times. While there are approximation techniques (i.e. backstepping) to determine the switching times, the condition is not easy to impose algorithmically. To develop a computationally tractable algorithm, [Papachristodoulou and Prajna, 2009] introduced the following theorem:

Theorem 2. Consider a hybrid system $\mathcal{H}$ defined in Definition 1 where each D_q is a subset of $\mathbb{R}^n$. Let x_e be a common equilibrium point in all discrete modes. If for all $q \in \mathcal{Q}$, let $V_q : D_q \to \mathbb{R}$ be a C^1 function such that:

(1) $V_q(x_e) = 0$ and $V_q(x) > 0$, $\forall x \in D_q \setminus \{x_e\}, q \in \mathcal{Q}$
(2) $\frac{\partial V_q}{\partial x} f_q(x) \leq 0$, $\forall x \in D_q, q \in \mathcal{Q}$
(3) $V_{q'}(R_{(q,q')}(x)) \leq V_q(x)$, $\forall x \in \{x | G_{(q,q')}(x) = 0\}$.

Then x_e is a stable equilibrium point of $\mathcal{H}$.

These Lyapunov theorems for hybrid systems either require the switching times of the system to be known or require decresence over the entire guard.

2.3 S-Procedure

In this section, we introduce the S-procedure, which is used to encode Lyapunov set containment conditions as an SOS program. The S-procedure is a special case of the Positivestellenstaz theorem with only inequality constraints [Jarvis-Wloszek et al., 2005].

Lemma 1. (S-procedure). Given $\{g_i\}_{i=1}^m \in \mathcal{R}[x]$. If there exist $\{s_i\}_{i=1}^m : \mathbb{R}^n \to \mathbb{R}_+$ such that
$-g_0(x) + \sum_{i=1}^m s_i(x) g_i(x) \geq 0$, then
$\bigcap_{i=1}^m \{x \in \mathbb{R}^n | g_i(x) \leq 0\} \subseteq \{x \in \mathbb{R}^n | g_0(x) \leq 0\}$.

To verify this, for all $i = [1, \ldots, m]$, an arbitrary x such that $g_i(x) \leq 0$ and positive $s_i(x)$'s, it must hold that $g_0(x) \leq 0$.

We extend the S-procedure to include equality constraints.

Lemma 2. (Extended S-procedure). Given $\{g_i\}_i^m \in \mathcal{R}[x]$ and $h \in \mathcal{R}[x]$. If there exist $\{s_i\}_i^m : \mathbb{R}^n \to \mathbb{R}_+$ and $r : \mathbb{R}^n \to \mathbb{R}$ such that $-g_0(x) + \sum_{i=1}^m s_i(x) g_i(x) + r(x) h(x) \geq 0$, then $\bigcap_{i=1}^m \{x \in \mathbb{R}^n | g_i(x) \leq 0\} \cap \{x \in R^n | h(x) = 0\} \subseteq \{x \in \mathbb{R}^n | g_0(x) \leq 0\}$.

To verify this, for all x such that $h(x) = 0$, this reduces to Lemma 1. By adding the polynomial r, the constraint matters only when $h(x) = 0$. When $h(x) \neq 0$, $r(x)$ can be set to a value to make the entire equation positive.

2.4 Sum of Squares Programming

SOS refers to way of representing a given polynomial in the form:

$$p = \sum_{i=1}^{N} g_i^2 \tag{1}$$

where $\{g_i\}_i^N \in \mathcal{R}[x]$ is a set of polynomials . If a polynomial is SOS, it is non-negative everywhere. We can check if a polynomial is SOS using semidefinite rogramming (SDP) solvers [Prajna et al., 2004, Lofberg, 2009]. It is important to note that requiring a polynomial to be SOS is more restrictive than just requiring positivity.

We can encode the S-procedures in Lemmas 1 and 2 by making s, r and the inequality be SOS. For example, a continuous time dynamical system with vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ and equilibrium point x_e, the local Lyapunov conditions are [Vidyasagar, 1993, Lemma 40]:

$$\begin{aligned} V(x_e) &= 0 \\ V(x) &> 0 \quad \forall x \in U \setminus \{x_e\} \\ \frac{\partial V}{\partial x} f(x) &\le 0 \quad \forall x \in U \end{aligned} \tag{2}$$

where U is a level set of V and inner-approximation of the ROA of f. This can be written using the S-procedure as:

$$-\frac{\partial V}{\partial x} f(x) + s(x)(V(x) - \gamma) \in \Sigma[x] \tag{3}$$

where $s \in \Sigma[x]$ and γ is the level set of V which defines U. This can be formulated as an LMI feasibility problem and solved with SDP solvers [Boyd and Vandenberghe, 1994].

3. STABILITY ANALYSIS

In this section, we present a local Lyapunov theorem and formulate it as an SOS optimization problem.

3.1 Problem Formulation

Instead of requiring decresence over the entire guard or knowledge of the switching times, we ensure that stable regions in mode q reset into the ROA of the next mode q'.

Theorem 3. Consider a hybrid system $\mathcal{H}$ defined in Definition 1. For all q, let U_q be a set in D_q and let U be the set defined by $U = \bigcup_q U_q$. Given an equilibrium point $x_e \in U$ in mode $q^\star$. For all $q \in \mathcal{Q} \setminus \{q^\star\}$, suppose there exist functions $V_q : D_q \to \mathbb{R}$ such that:

(1) $V_q(x) > 0 \quad \forall x \in U_q$
(2) $\frac{\partial V_q}{\partial x} f_q(x) \le 0 \quad \forall \text{q s.t. } x \in U_q$
(3) $R_{(q,q')}(x) \in U_{q'} \quad \forall (q,q')$ s.t. $x \in U_q$ and $G_{(q,q')}(x) = 0$

and for $q^\star$

(1) $V_{q^\star}(x) > 0 \quad \forall x \in U_q^\star$
(2) $V_{q^\star}(x_e) = 0$
(3) $\frac{\partial V_{q^\star}}{\partial x} f_{q^\star}(x) \le 0 \quad \forall x \in U_{q^\star}$

then U is an invariant set. Moreover, if $\frac{\partial V_q}{\partial x} f_q(x) < 0 \quad \forall (q,x) \in U$ with all trajectories not visiting a discrete state more than once, then $(q^\star, x_e)$ is locally asymptotically stable in U.

Proof 4. Consider a hybrid system with a limit cycle transversing three discrete modes $\{1,2,3\}$ with guards $\{G_{(q_1,q_3)}, G_{(q_2,q_1)}, G_{(q_3,q_2)}\}$. U_{q_1} is either invariant (by the standard Lyapunov proof [Khalil, 2002]) or contains trajectories that reset into U_{q_3}, by condition 3. Likewise, U_{q_3} is either invariant or contains trajectories that reset into U_{q_2} and U_{q_2} is either invariant or contains trajectories that reset into U_{q_1}. Therefore, $U_{q_1} \cup U_{q_2} \cup U_{q_3}$ is invariant. This extends to systems with 2 or more discrete modes.

Assume $\frac{\partial V_q}{\partial x} f_q(x) < 0 \quad \forall (q,x) \in U$ with all trajectories not visiting a discrete state more than once. Consider a hybrid system with two modes with $G_{(q_2,q_1)}$ and let q_1 be the discrete state with equilibrium point x_e. By the proof in [Khalil, 2002], U_{q_1} is invariant. Now, we prove that trajectories in U_{q_2} reset into U_{q_1}. Assume not. Let x_0 be an initial condition of $\mathcal{H}$ starting in U_{q_2}. Since $\frac{\partial V_2}{\partial x} f_2(x) < 0$, there exists a $T \in \mathbb{R}_+$ such that $V_2(x(T)) = V_2(x_0) + \int_0^T \frac{\partial V_q}{\partial x} f_2(x) dx = 0$. However, this contradicts the assumption that $V_2(x) > 0$ for all $x \in U_{q_2}$. Therefore, the trajectory must reach guard $g_{1,2}$. Therefore, $U_{q_1} \cup U_{q_2}$ is invariant. This extends to more discrete states. □

3.2 SOS Formulation

Next, we show how to encode Theorem 3 as a set containment problem. To make this computationally feasible, we set U_q to be the γ_q level set of V_q.

We can formulate our Lyapunov conditions from Theorem 3 as a set containment. The ROA of the system is taken to be every point contained in the set $\bigcup_{q \in \mathcal{Q}} \{x \in D_q | V_q(x) \le \gamma_q\}$. Our candidate Lyapunov functions are only required to be valid over the set of states where our guards are valid (positive) i.e. $\bigcap_{(q,q') \in \mathcal{E}} \{x \in D_q | G_{(q,q')}(x) \ge 0\}$. For the γ_q-level set to be the ROA for discrete mode q, the intersection of the γ_q-level set and D_q must lie within the set of states that has a negative Lie derivative $\{x \in D_q | \frac{\partial V_q}{\partial x} f_q(x) < 0\}$. This gives the set containment:

$$\begin{aligned} \bigcap_{(q,q') \in \mathcal{E}} \{x \in D_q | G_{(q,q')}(x) \ge 0\} \bigcap \{x \in D_q | V_q(x) \le \gamma_q\} \\ \subset \{x \in D_q | \frac{\partial V_q}{\partial x} f_q(x) < 0\} \quad \forall q \in \mathcal{Q} \end{aligned} \tag{4}$$

Similarly we look at the reset condition between two modes. Consider two modes q and q' such that mode q transitions from q to q' along guard $G_{(q,q')}$ with the reset map $R_{(q,q')}$. We require that the states that are in the ROA of mode q that intersect the guard $g_{1,2}$ reset inside the ROA of mode q'. This can be written as the set containment:

$$\begin{aligned} \{x \in D_q | x \in G_{(q,q')} = 0\} \cap \{x \in D_q | V_q(x) \le \gamma_q\} \\ \subset \{x \in D_q | V_{q'}(R_{(q,q')}(x)) \le \gamma_{q'}\} \quad \forall (q,q') \in \mathcal{E} \end{aligned} \tag{5}$$

From the set containment problem, we use the S-procedure in Lemmas 1 and 2 to obtain the SOS optimization program.

To incorporate the fact that V_q only needs to be positive on D_q, we impose the relaxed condition found in [Papachristodoulou and Prajna, 2002]:

$$V_q(x) - \sum_{q' \ne q} G_{(q,q')}(x) t_{q,q'} > 0 \tag{6}$$

where $t_{q,q'}$ is a SOS polynomial.

Using the S-procedure in Lemma 1, we convert the Lyapunov conditions (4) to:

$$-\frac{\partial V_q}{\partial x} f_q - (\gamma_q - V_q(x)) s_{q,2} - \sum_{q' \neq q} G_{(q,q')}(x) w_{q,q'} \in \Sigma[x] \tag{7}$$

where $s_{q,2}, w_{q,q'} \in \Sigma[x]$.

Using the extended S-procedure in Lemma 2, we convert the guard condition (5) to:

$$(V_q - \gamma_q) y_{q,q'} + (\gamma_{q'} - V_{q'}(R_e(x))) z_{q,q'} + G_{ij} r_{q,q'} > \Sigma[x] \tag{8}$$

where $y_{q,q'}, z_{q,q'} \in \Sigma[x]$ and $r_{q,q'} \in \mathcal{R}[x]$. By allowing $p_{g,q,4,q'}$ to be any polynomial, we ensure that the guard condition is only satisfied when the system hits the guard, not before or after.

To provide a measurement of growth for the ROA of a system, we use a shape function $p_q : \mathbb{R}^n \to \mathbb{R}$ for all $q \in \mathcal{Q}$ introduced in [Tan, 2008]. This provides an absolute scale to measure the size of the ROA. Otherwise, due to the scaling of V_q, γ_q can be arbitrarily large. The shape function is chosen to reflect the relative importance of the states and must be chosen appropriately. This constraint is formulated as such:

$$\{x | p_q(x) \leq \beta_q\} \subset \{x | V_q(x) \leq \gamma_q\} \quad \forall q \in \mathcal{Q} \tag{9}$$

This can be formulated as a SOS constraint as:

$$(\gamma_q - V_q) - (\beta_q - p_q) s_{q,3} \in \Sigma[x] \quad \forall q \in \mathcal{Q} \tag{10}$$

Corollary 1. As a shorthand, we refer to $\{t_{q,q'}, s_{q,2}, s_{i,3}, w_{q,q'}, y_{q,q'}, z_{q,q'}\}_{(q,q) \in \mathcal{E}}$ as s, and $\{r_{q,q'}\}_{(q,q) \in \mathcal{E}}$ as r. Consider a hybrid system, assume there exist $\{V_q\}_{q \in \mathcal{Q}}$ and $s \in \Sigma[x]$, $r \in \mathcal{R}[x]$, $\gamma_q, \beta_q \in \mathbb{R}\, \forall (q, q') \in \mathcal{E}$ such that

$$\begin{aligned}
&V_q(x) - \sum_{q' \neq q} G_{(q,q')}(x) t_{q,q'} \in \Sigma[x] \\
&-\frac{\partial V_q}{\partial x} f_q - (\gamma_q - V_q(x)) s_{q,2} - \sum_{q' \neq q} G_{(q,q')}(x) w_{q,q'} \in \Sigma[x] \\
&(\gamma_q - V_q) - (\beta_q - p_q) s_{q,3} \in \Sigma[x] \\
&(V_q - \gamma_q) y_{q,q'} + (\gamma_{q'} - V_{q'}(R_e(x))) z_{q,q'} \\
&\qquad + G_{(q,q')} r_{q,q'} \in \Sigma[x] \quad \forall j \text{ s.t. } (q, q') \in \mathcal{E}
\end{aligned} \tag{11}$$

Then the origin is locally asymptotically stable with $\bigcup_q \{x | x \in D_q \text{ and } V_q(x) \leq \gamma_q\} \subset A$.

This corollary follows from combining Theorem 3 and Equations (6), (7), (8) and (10) and convert the positivity constraints to SOS constraints.

Using Corollary 1, we write the optimization problem to determine the ROA.

$$\begin{aligned}
&\max_{V, s \in \Sigma[x], h \in \mathcal{R}[x]} \sum_{q \in \mathcal{Q}} \beta_q \\
&\text{such that} \\
&V_q(x) - \sum_{q' \neq q} G_{(q,q')}(x) t_{q,q'} \in \Sigma[x] \\
&-\frac{\partial V_q}{\partial x} f_q - (\gamma_q - V_q(x)) s_{q,2} - \sum_{q' \neq q} G_{(q,q')}(x) w_{q,q'} \in \Sigma[x] \\
&(\gamma_q - V_q) - (\beta_q - p_q) s_{q,3} \in \Sigma[x] \\
&(V_q - \gamma_q) y_{q,q'} + (\gamma_{q'} - V_{q'}(R_e(x))) z_{q,q'} \\
&\qquad + G_{(q,q')} r_{q,q'} \in \Sigma[x] \quad \forall j \text{ s.t. } (q, q') \in \mathcal{E}
\end{aligned} \tag{12}$$

The SOS constraints in this optimization can be converted into SDP constraints (as discussed in Section 2.4). As this optimization problem is bilinear in the decision variables s, p and V_q, we solve this by fixing V_q and solving for s, p; then fix s, p and solve for V_q. Further details can be found in [Packard et al., 2009].

This optimization problem avoids having prior knowledge of the switching times or stability across the entire guard by dealing with set containments across the guard and reset maps. Due to SOS programming, we are limited to polynomial reset maps and Lyapunov functions without a constant term.

4. MODELING TECHNIQUES

In this section we introduce the tools and techniques we found helpful for approximating the ROA of a system using SOS techniques.

4.1 Changing Coordinates

When modeling a system, there are often multiple ways of choosing a descriptive set of states. While these different models may be equivalent, a careful choice of variables can reduce the intractability of certain problems. One of the most restrictive problems we found were conditions on the discrete dynamics between hybrid modes. When searching for a ROA over several hybrid modes, we found that having reset maps with non-constant terms resulted in larger computed ROA level sets, and allowed intractable problems to be solved (Section 5.5). This can be done by rewriting the dynamics of the system using a different choice of variables so that the reset map does not have constant terms.

4.2 Guards

While guards are typically components of hybrid systems, it is possible to use guards to reduce the state space that the Lyapunov-like function is required to be valid. Our SOS formulation only requires the Lyapunov-like conditions on V to be valid over the set of states that satisfy our guards[Papachristodoulou and Prajna, 2002]. This property can be used to remove regions of the state space that have dynamics that limit the growth of the region of attraction by allowing the ROA to violate the properties of decresence etc. when outside the valid domain.

4.3 Constraints

In contrast to guards, constraints ensure that a reachable set stays within a desired state space, forcing the ROA to always satisfy a set of given constraints. We present this as a method of modeling physical constraints on the system (the position of the object must stay above the ground, the shock absorber cannot travel more than n cm etc.). However, this is also a useful tool for bounding the state space to allow for faster computation times. In the case of globally stable systems, the ROA grows unbounded, resulting in an error during optimization. By constraining the ROA to stay within a subset of our space, this unbounded growth can be limited.

Constraints can be written as follows. Let $\{C_k\}_{k=0}^{n_k} : D \to \mathbb{R}$ be the functions that constrain the state space of the system. This restricts D to $\bigcap_{k=0}^{n_k}\{x|C_k(x) \geq 0\}$. We add the set containment constraint: for all $k = 1, \ldots, n_k$,

$$\bigcap_{(q,q')\in\mathcal{E}} \{x \in D|G_{(q,q')}(x) \geq 0\} \bigcap \{x \in D|V_q(x) \leq \gamma_q\} \subset \{x \in D|C_k(x) \geq 0\} \quad \forall q \in \mathcal{Q} \tag{13}$$

and use the S-procedure to obtain the SOS constraint.

4.4 Seeding

The initial 'seed' candidates for V_q, γ_q and β_q must be feasible for the optimization to begin. Small initial values for γ_q are commonly used to allow for the initial shape function to be the Lyapunov function for the linearized system. A problem arises when the ROA is highly dependent on repeated discrete transitions (Examples 5.4, and 5.5). Using small initial values would result in negligible growth as the image of the reset map limits the size of the domain. To overcome this, a large γ value is initially used for the domain of the next mode, allowing the domain to grow under the assumption the ROA in the next mode is the entire space. When the ROA of the domain of the next mode is computed, this assumption can be revoked by reinitializing γ to a small initial value.

4.5 Scaling

When optimizing over state variables, it is useful to keep their relative order of magnitude similar in size. This prevents certain states 'overpowering' others during optimization. It is also useful to avoid state variables from becoming either too large or to small which can lead to inaccuracies due to machine precision. We found that this resulted in a faster computation times.

We do this by modifying Equation (7) by adding a scaling term $s_{q,1}$:

$$-\frac{\partial V_q}{\partial x} f_q s_{q,1} - (\gamma_q - V_q(x))s_{q,2} - \sum_{q'\neq q} G_{(q,q')}(x) w_{q,q'} \in \Sigma[x] \tag{14}$$

where $s_{q,1}, s_{q,2}, w_{q,q'} \in \Sigma[x]$.

4.6 Bases and Degree for the SOS variables

A careful choice of both the monomial bases and degree of the SOS variables is important for both computation time as well as the final size of the ROA. Searching over a higher degree Lyapunov function increases computation time but may be able to determine a larger volume ROA and improve problem tractability. The complexity of a problem can be reduced by carefully selecting the base monomials for the s variables.

4.7 Sweeping Lyapunov functions

The optimization problem in Equation (12) grows the ROA with respect to the shape functions p_q. To obtain a larger ROA, we perform the optimization problem for various shape functions p to 'sweep' out the state space. As seen in Figure 1, a narrow shape function p can be used to search over a particular combination of states. The union of the computed ROAs offer a better approximation of the true ROA of the system.

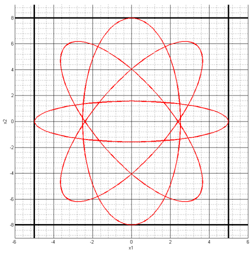


Fig. 1: Overlaying several ROAs found using the sweeping method. Lyapunov functions are shown in red, and the true ROA is outlined in black.

4.8 Ensuring SOS

When performing each optimization step it is important to check that the SOS constraints are satisfied by the returned solutions. Depending on the SOS package and the solver, the solutions that were reported as feasible may turn out to be SOS to machine precision. To verify a returned SOS variable, we perform post computation cleaning and checking. First, we replace any "small" coefficients in the SOS variables with zeros, remove SOS variables that are only SOS due to computational accuracy. The variable is then converted into a matrix where different matrix decompositions (Cholesky, EVD etc.) were used to ensure the positive definiteness of the resulting matrix. If the matrix is found to be positive definite after cleaning, we then use the zeroed matrix as our SOS solution. This process is repeated for every SOS variable. If any SOS variable is not SOS, then that solution step is discarded and repeated.

4.9 Slack

By adding a slack variable into our constraint we can ensure that the computed ROA returned is within the actual ROA of the system. Without this, the computed ROA can overshoot the actual ROA due to rounding errors when low order terms are removed. We can explicitly add a slack term in the γ-step. Rather than finding SOS variables such that the SOS problem presented in Equation (12) is feasible, we instead try to minimize our slack variable S under the conditions:

$$\begin{aligned} -\frac{\partial V_q}{\partial x} f_q s_{q,1} - (\gamma - V_q(x))s_{q,2} \\ - \sum_{q'\neq q} G_{q,q'}(x) w_{q,q'} - S \in \Sigma \\ S > 0.99999 S_{old} \end{aligned} \tag{15}$$

This process requires that the returned solution is close to the previously computed solution [Lofberg, 2011].

5. EXAMPLES

In this section we cover a number of examples that highlight important aspects and results of SOS modeling and our framework. We implement these algorithms using YALMIP and Mosek. All examples were run on a computer with a Intel i5-2320 CPU running at 3GHz and 16Gb of RAM.

In the following analysis we represent our hybrid systems using the graphical notation seen in Figure 2.

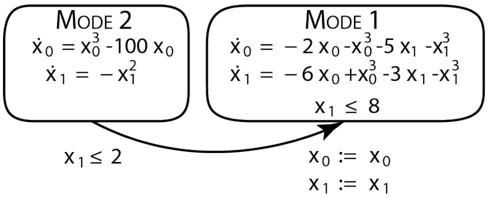


Fig. 2. Cartoon for the non-linear system. Mode 1 is globally stable about the origin, while mode 2 is unstable in x_1 and is only locally stable in x_0.

For clarity, we describe the features of the hybrid system as illustrated in this figure. We represent different hybrid modes by the rounded rectangles, with their name written prominently inside. The continuous dynamics are written inside each rectangle directly under the name of the mode. At the base of the rectangle we write any constraints imposed on the system. In this way, we can see mode 1 has been constrained so that $x_1 \leq 8$. The discrete transitions are indicated by an arrow indicating the direction of the transition. At the tail of the arrow, we write the reset condition, and at the tip we write the reset map. Looking at the discrete transition from mode 2 to mode 1, we see that whenever a trajectory satisfies the condition $x_1 \leq 2$, it undergoes a discrete transition to mode 1. The states are updated based on the update rule seen at the tip of the arrow. In our example, we have a trivial reset map, x_0 and x_1 in mode 1 are taken to be the x_0 and x_1 in mode 2 respectively at the transition point.

A more complicated reset map can be seen in the Lunar Lander example which we introduce in Section 5.5. Looking at the reset map from mode 2 to mode 1 in Figure 11, we can see that the discrete reset a linear function. In this example, state x_0 in mode 1 is found by taking the values of x_0 and x_2 in mode 2 and performing the update $x_0 + \frac{m_0}{m_0+m_2} x_2$.

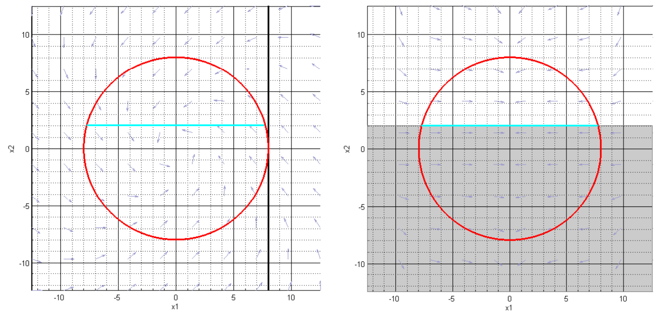
5.1 Non-linear Systems and Partially Stable Guards

The first system we look at has two non-linear modes and a partially stable guard Figure 2. The computed ROA for this system are shown in Figure 3. The Lyapunov-like function in mode 1 takes its expected form of a circle of radius 8 due to the constraint. Based on the dynamics of mode 2, it is expected that the ROA would grow to be a circle of radius 10. The discrete transition however constrains this growth as the image of the guard must lie within the ROA of mode 1. This leads to the ROA in mode 2 being confined by its image in mode 1.

5.2 Interaction of Two Guards

The next system is shown in Figure 4. This system consists of three modes. Mode 1 is globally stable, and mode 3 is unstable. Mode 2 has two discrete guards, transitioning to mode 1 and mode 3 respectively.

The computed ROA for the hybrid system is shown in Figure 5. Mode 1 is globally stable, as seen by the ROA expanding up to the edge of its constraint. The discrete guard in mode 2 is locally stable. The active subset of the discrete transition is highlighted in cyan. Looking at the image of this guard in mode 1, it may be expected that the ROA in mode 2 could grow larger. However looking in mode 2, we see that this is not the case. The second discrete transition in mode 2 is globally unstable, limiting the ROA of mode 2 to lie to the left of the line $x_0 = 1$.



(a) Mode 1 (b) Mode 2

Fig. 3. Computed ROA for the hybrid system illustrated in Figure 2. ROA is shown in red, constraints in black and guards are indicated by the shaded gray region.

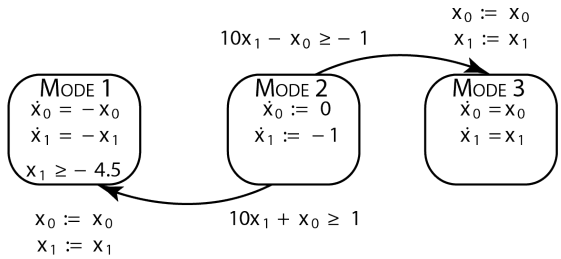
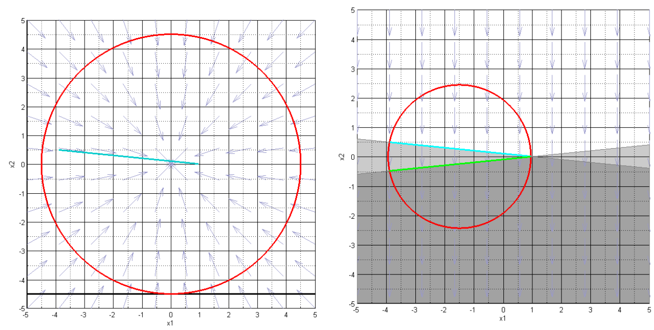


Fig. 4. Hybrid model for our Two Guard system. Continuous dynamics are written inside each mode and discrete dynamics are indicated by the arrows with the guard condition on the tail and the reset map on the tip.



(a) Mode 1 (b) Mode 2

Fig. 5. Two of the three modes for our two guard system with ROA indicated in red. Discrete transitions from mode 2 to mode 1 and mode 2 to mode 3 are indicated by the cyan and green lines respectively.

5.3 Changes in dimension

We introduce a hybrid system that can expand and contract in dimension between each mode as shown in Figure 6. This system produces the ROA seen in Figure 7. As illustrated, the ROA expands clearly in each mode showing that systems of varying dimension can be analyzed through SOS methods.

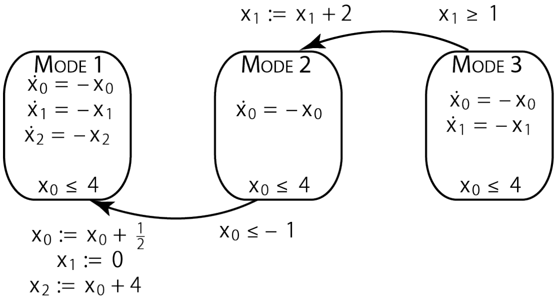
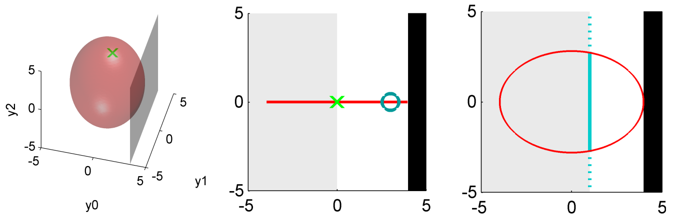


Fig. 6. Hybrid model for our varying dimension system.



(a) Mode 1 (b) Mode 2 (c) Mode 3

Fig. 7. Computed ROA for the hybrid system illustrated in Figure 6. Discrete transitions from mode 2 to mode 1 is indicated by the green line pair and transitions from mode 3 to mode 2 are shown as the cyan pair.

5.4 Bouncing Ball

One of the canonical examples for hybrid systems is the bouncing ball- a system whose equilibrium only exists due to an infinite number of discrete transitions in the system. These systems can exhibit *Zeno* behavior whereby an infinite number of discrete transitions can occur in finite time.

Consider the system shown in Figure 8. Here we have a ball modeled as a point mass, falling under gravity and resistive forces that are proportional to velocity. We impose a constraint on the maximum velocity to prevent the system from becoming unbounded. Each time the ball collides, the time between collisions decreases due to energy loss in the system. In this manner, Zeno behavior is observed- over a finite amount of time, the system can have an infinite number of discrete transitions.

Figure 9 shows the computed ROA for this system as well as two sample trajectories. Following these two trajectories, we can see that the system is indeed stable and performs an infinite number of smaller and smaller transitions over finite time.

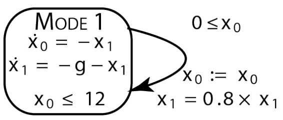


Fig. 8. Hybrid model for the bouncing ball system. The Discrete transition maps states from mode 1, back into mode 1 representing the near instantaneous change in direction caused by the bounce.

5.5 Lunar Lander

Our final example is that of a moon lander, a variant of the mass-spring vertical hopper system. Consider the system

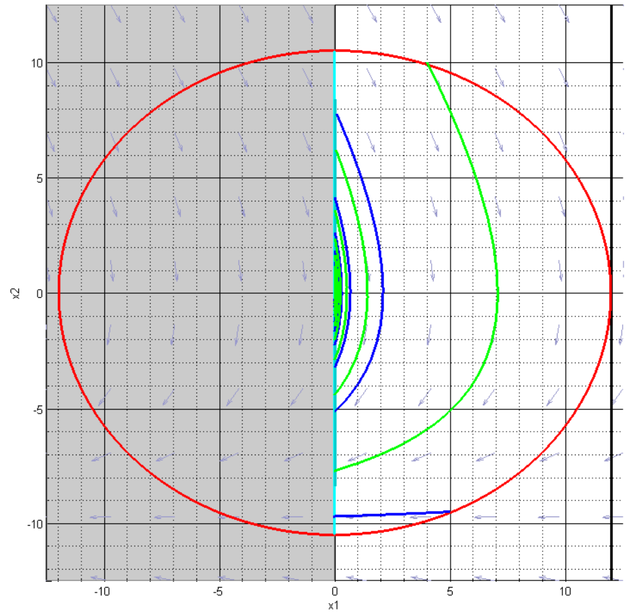


Fig. 9. Computed ROA for the hybrid system illustrated in Figure 8. The ROA is shown in red, and the guard and its image are shown in cyan and dark cyan respectively. Two system trajectories are shown in green and blue.

seen in Figure 11. The system consists of two masses that are coupled by a spring and a damper which make up the shock absorber system. The shock absorber has a physical length limit, and the total length cannot exceed 0.8m in either direction. These masses have two inputs that can be used to apply a force on the two masses. Our design problem is to find a ROA for the system given a control law such that the such that the shock absorber never exceeds its maximum length.

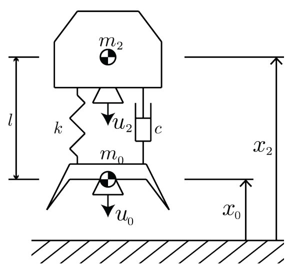


Fig. 10. Diagram of the Lunar Lander system. Coordinate frames, location of the centers of mass, spring and damper couplings and thrusters are all indicated.

The system consists of two modes, an aerial mode and a ground mode. These two modes have different continuous dynamics, with the ground mode consisting of purely a mass-spring-damper system, while the aerial mode also includes the falling dynamics. Our hybrid transitions are taken to be when the lower mass touches the ground and when the forces on the lower mass change sign, causing the foot to rise. The dynamics were chosen to provide a polynomial reset map without a constant term.

Figure 12 shows several important features that can be used for analysis of our system. First, it sets limitations on

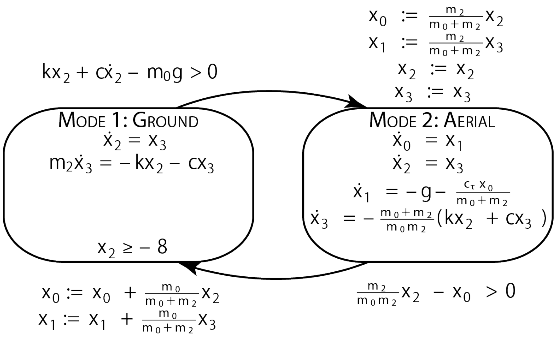
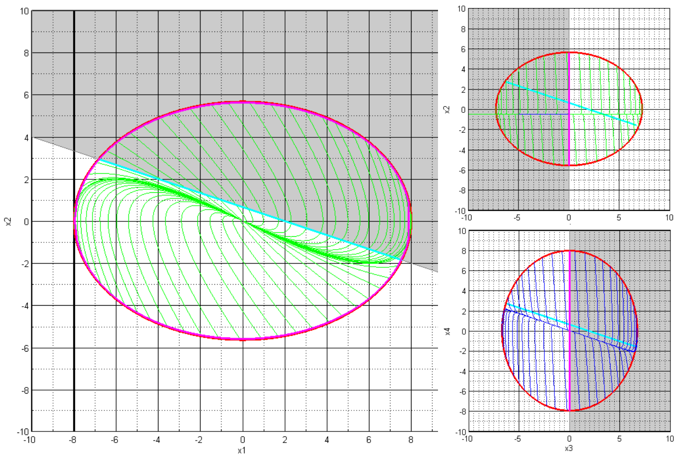


Fig. 11. Hybrid model of the Lunar Lander system



(a) Mode 1 (b) Mode 2

Fig. 12. ROA (red) for the Lunar Lander. Left: Phase diagram of the Ground mode with the state being the variable $\bar{x}_2$ which represents the offset of the top mass from the mass center, Right: Phase diagram of the Aerial mode with the states being the height of the mass center above the ground $\bar{x}_1$ (top) and the deviation from the mass center of the top mass $\bar{x}_2$ (bottom). Discrete transitions from mode 2 to mode 1 are shown by the pink plane in mode 2 to the pink oval in mode 1 (overlapping the ROA in mode 1 completely), and the discrete transition from mode 1 to mode 2 is the cyan line in mode 1 transitioning to the cyan hypersurface in mode 2.

the airborne speed and positions that will result in a safe landing that does not damage the landing gear. Second, it shows that the system has a stable equilibrium point in the ground mode where it is at rest.

6. CONCLUSION

We present a theorem and numerical method for stability analysis of hybrid systems that can handle stability over only a subset of the guard and does not require prior knowledge of switching times and common equilibrium among all the discrete modes. The method is based on polynomial and piecewise polynomial Lyapunov functions. We use SOS programming to find the level sets of multiple Lyapunov functions to find local ROAs for autonomous hybrid systems without input. We also list many methods which were implemented to get the SOS program to run find the ROA for several examples.

There are a number of limitations with using the SOS approach for finding the ROA of a system. These problems arise through issues in problem formulation as well as the solvers being used.

When formulating an SOS problem, we are constrained to solving polynomial systems. This means that the continuous and discrete dynamics of the system need to be represented in polynomial form which may cause approximation errors. We found that reset maps could not have any constant terms. For systems such as Example 5.5, this can be avoided through coordinate changes, adding additional complexity. The initial conditions of a system need to be valid for the system to grow. While this can be simple for trivial systems, it can be more challenging for more complicated, real-world systems, especially systems that only exhibit stability through discrete switching. This is a major limitation as it often requires the initial shape function to be chosen intelligently before the SOS procedure can be started.

There are also challenges in solving these SOS problems. To solve an SOS problem, an SDP solver needs to be used. SDP problems are challenging which can lead to computation errors. This can result in SOS polynomials being returned that are not positive and solvers only providing valid results for certain numbers of guards and constraints. Our formulation is bilinear in the decision variables, requiring multiple iterations for a solution and may converge to a local minimum. The solution given by a solver is highly dependent on the order of the SOS variables that are being looked over. If a problem requires a high order Taylor expansion, the combination of large order SOS variables combined with high ordered dynamics can lead to a prohibitively large problem that can take hours to complete each step.

Future work includes determining how to intelligently chose the shape function and the degrees of the slack variables. Given more powerful SDP solvers, these limitations may be mitigated, but presently they pose a significant challenge when trying to find the ROA using SOS methods. Recent methods using occupation measures [Henrion and Lasserre, 2012, Henrion and Korda, 2013, Majumdar et al., 2013] may be used to eliminate the bilinearity of the optimization problem.

ACKNOWLEDGEMENTS

This work was supported by Department of Defense Office of Naval Research Grant N00014-13-1-0341 and National Science Foundation Grant 0903711 and EAGER Grant 31963-23797-44-EHR4B

REFERENCES

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 1994. ISBN 0521833787.

Michael S Branicky. Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, April 1998. ISSN 00189286. doi: 10.1109/9.664150.

M.S. Branicky. Stability of Switched and Hybrid Systems. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 4, pages 3498–3503. IEEE, 1994.

Chaohong Cai, Andrew R. Teel, and Rafal Goebel. Smooth Lyapunov Functions for Hybrid Systems Part II: (Pre)Asymptotically Stable Compact Sets. *IEEE Transactions on Automatic Control*, 53(3):734–748, April 2008. ISSN 0018-9286. doi: 10.1109/TAC.2008.919257.

Rafal Goebel and Andrew R. Teel. Preasymptotic Stability and Homogeneous Approximations of Hybrid Dynamical Systems. *SIAM Review*, 52(1):87–109, January 2010. ISSN 0036-1445. doi: 10.1137/090749797.

Didier Henrion and Milan Korda. Convex Computation of the Region of Attraction of Polynomial Control Systems. *IEEE Transactions on Automatic Control*, pages 1–1, 2013. ISSN 0018-9286. doi: 10.1109/TAC.2013.2283095.

Didier Henrion and Jean-Bernard Lasserre. Inner Approximations for Polynomial Matrix Inequalities and Robust Stability Regions. *IEEE Transactions on Automatic Control*, 57(6):1456–1467, June 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2178717.

Zachary Jarvis-Wloszek, R. Feeley, Weehong Tan, K. Sun, and Andrew Packard. Control Applications of Sum of Squares Programming. *Positive Polynomials in Control*, pages 580–580, 2005. doi: 10.1007/10997703_1.

M. Johansson and Anders Rantzer. Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems. *IEEE Transactions on Automatic Control*, 43(4):555–559, April 1998. ISSN 00189286. doi: 10.1109/9.664157.

Hassan Khalil. *Nonlinear Systems Third Edition*. Prentice Hall, 2002. ISBN 0130673897.

Liberzon and Daniel. *Switching in Systems and Control*. Springer, 2003.

D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999. ISSN 02721708.

Johan Lofberg. YALMIP : a Toolbox for Modeling and Optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pages 284–289. IEEE, 2004. ISBN 0-7803-8636-1. doi: 10.1109/CACSD.2004.1393890.

Johan Lofberg. Pre- and Post-Processing Sum-of-Squares Programs in Practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011, May 2009. ISSN 0018-9286. doi: 10.1109/TAC.2009.2017144.

Johan Lofberg. Strictly Feasible Sum-of-Squares Solutions, 2011.

Anirudha Majumdar, Amir Ali Ahmadi, and Russ Tedrake. Control Design along Trajectories with Sums of Squares Programming. October 2012.

Anirudha Majumdar, Ram Vasudevan, MM Tobenkin, and Russ Tedrake. Convex Optimization of Nonlinear Feedback Controllers via Occupation Measures. 2013.

Ian R. Manchester, Mark M. Tobenkin, Michael Levashov, and Russ Tedrake. Regions of Attraction for Hybrid Limit Cycles of Walking Robots. In *Arxiv preprint arXiv:1010.2247*, pages 1–8, 2010.

MOSEK. The MOSEK Optimization Tools Version 2.5. Users Manual and Reference, 2002, 2002.

Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, June 1987. ISSN 0025-5610.

Andrew Packard, Peter Seiler, and Gary Balas. Quantitative Local Analysis of Nonlinear Systems (Short course notes), 2009.

Antonis Papachristodoulou and Stephen Prajna. On the Construction of Lyapunov Functions Using the Sum of Squares Decomposition. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 3(1):3482–3487, 2002. doi: 10.1109/CDC.2002.1184414.

Antonis Papachristodoulou and Stephen Prajna. A Tutorial on Sum of Squares Techniques for Systems Analysis. *Proceedings of the 2005, American Control Conference, 2005.*, pages 2686–2700, 2005. doi: 10.1109/ACC.2005.1470374.

Antonis Papachristodoulou and Stephen Prajna. Robust Stability Analysis of Nonlinear Hybrid Systems. *IEEE Transactions on Automatic Control*, 54 (5):1035–1041, May 2009. ISSN 0018-9286. doi: 10.1109/TAC.2009.2017155.

Pablo A Parrilo. *Structured Semide nite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, 2000.

Michael Posa, Mark M. Tobenkin, and Russ Tedrake. Lyapunov Analysis of Rigid Body Systems with Impacts and Friction via Sums-of-Squares. In *Proceedings of the 16th international conference on Hybrid systems: computation and control - HSCC '13*, page 63, New York, New York, USA, April 2013. ACM Press.

Stephen Prajna and Anders Rantzer. Convex Programs for Temporal Verification of Nonlinear Dynamical Systems. *SIAM Journal on Control and Optimization*, 46(3):999–1021, January 2007. ISSN 0363-0129.

Stephen Prajna, Antonis Papachristodoulou, Peter Seiler, and Pablo A Parrilo. *SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB*, 2004.

Shankar Sastry. *Nonlinear Systems*. Springer, 1999.

Robert Shorten, Fabian Wirth, Oliver Mason, Kai Wulff, and Christopher King. Stability Criteria for Switched and Hybrid Systems. *SIAM Review*, 49(4):545–592, January 2007. ISSN 0036-1445. doi: 10.1137/05063516X.

J. F. Sturm. Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software,*, 11(12):625–653, 1999.

Weehong Tan. Stability Region Analysis using Polynomial and Composite Polynomial Lyapunov Functions and Sum-of-Squares Programming. *Automatic Control, IEEE Transactions on*, 53(2):565–571, 2008.

Mathukumalli Vidyasagar. *Nonlinear System Analysis*. Prentice Hall, 1993.