Research Article

Exact Asymptotic Stability Analysis and Region-of-Attraction Estimation for Nonlinear Systems

Min Wu,[1] Zhengfeng Yang,[1] and Wang Lin[2]

[1] *Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China*
[2] *College of Mathematics and Information Science, Wenzhou University, Wenzhou Zhejiang 325035, China*

Correspondence should be addressed to Wang Lin; linwang@wzu.edu.cn

We address the problem of asymptotic stability and region-of-attraction analysis of nonlinear dynamical systems. A hybrid symbolic-numeric method is presented to compute exact Lyapunov functions and exact estimates of regions of attraction of nonlinear systems efficiently. A numerical Lyapunov function and an estimate of region of attraction can be obtained by solving an (bilinear) SOS programming via BMI solver, then the modified Newton refinement and rational vector recovery techniques are applied to obtain exact Lyapunov functions and verified estimates of regions of attraction with rational coefficients. Experiments on some benchmarks are given to illustrate the efficiency of our algorithm.

1. Introduction

Lyapunov's stability theory, which concern is with the behavior of trajectories of differential systems, plays an important role in analysis and synthesis of nonlinear continuous systems. Lyapunov functions can be used to verify the asymptotic stability, including locally asymptotic stability and globally asymptotic stability. In the literature, there has been lots of work on constructing Lyapunov functions [1–12]. For example, [4, 5] used the linear matrix inequality (LMI) method to compute quadratic Lyapunov functions. In [8, 13], based on sum of squares (SOS) decomposition, a method is proposed to construct high-degree numerical polynomial Lyapunov functions. Reference [9] proposed a new method for computing polynomials Lyapunov functions by solving semialgebraic constraints via the tool DISCOVERER [14]. Reference [15] constructed Lyapunov functions beyond polynomials by using radial basis functions. In [11], the Gröbner based method is used to choose the parameters in Lyapunov functions in an optimal way.

Since the analysis of asymptotic stability is not sufficient for safety critical systems, the analysis of the region of attraction (ROA) of an asymptotically stable equilibrium point is a topic of significant importance. The ROA is the set of initial states from which the system converges to the equilibrium point. Computing exact regions of attraction (ROAs) for nonlinear dynamical systems is very hard if not impossible; therefore, researchers have focused on finding estimates of the actual ROAs. There are many well-established techniques for computation of ROAs [5, 16–22]. Among all methods, those based on Lyapunov functions are dominant in the literature. These methods compute both a Lyapunov function as a stability certificate and the sublevel sets of this Lyapunov function, which give rise to estimates of the ROA. In [17], the authors computed quadratic Lyapunov functions to optimize the volume of an ellipsoid contained in ROA by using nonlinear programming. Reference [5] employed LMI based method to compute the optimal quadratic Lyapunov function for maximizing the volume of an ellipsoid contained in the ROA of odd polynomial systems. In [22], based on SOS decomposition, a method is presented to search for polynomial Lyapunov functions that enlarge a provable ROA of nonlinear polynomial system. Reference [18] used discretization (in time) to flow invariant sets backwards along the flow of the vector field, obtaining larger and larger estimates for the ROA.

Reference [21] employed quantifier elimination (QE) method via QEPCAD to find Lyapunov functions for estimating the ROA.

Taking advantage of the efficiency of numerical computation and the error-free property of symbolic computation, in this paper, we present a hybrid symbolic-numeric algorithm to compute exact Lyapunov functions for asymptotic stability and verified estimates of the ROAs of continuous systems. The algorithm is based on SOS relaxation [23] of a parametric polynomial optimization problem with LMI or bilinear matrix inequality (BMI) constraints, which can be solved directly using LMI or BMI solver in MAT-LAB, such as SOSTOOLS [24], YALMIP [25], SeDuMi [26], PENBMI [27], and exact SOS representation recovery techniques presented in [28, 29]. Unlike the numerical approaches, our method can yield exact Lyapunov functions and verified estimates of ROAs, which can overcome the unsoundness in analysis of nonlinear systems caused by numerical errors [30]. In comparison with some symbolic approaches based on qualifier elimination technique, our approach is more efficient and practical, because parametric polynomial optimization problem based on SOS relaxation with fixed size can be solved in polynomial time theoretically.

The rest of the paper is organized as follows. In Section 2, we introduce some definitions and notions about Lyapunov stability and ROA. Section 3 is devoted to transform the problem of computing Lyapunov functions and estimates of ROAs into a parametric program with LMI or BMI constraints. In Section 4, a symbolic-numeric approach via SOS relaxation and exact rational vector recovery is proposed to compute Lyapunov functions and estimates of ROAs, and an algorithm is described. In Section 5, experiments on some benchmarks are shown to illustrate our algorithm on asymptotic stability and ROA analysis. At last, we conclude our results in Section 6.

2. Lyapunov Stability and Region of Attraction

In this section, we will present the notion of Lyapunov stability and regions of attraction (ROAs) of dynamical systems.

Consider the autonomous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1}$$

where $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuous and $\mathbf{f}$ satisfies the Lipschitz condition. Denote by $\phi(t; \mathbf{x}_0)$ the solution of (1) corresponding to the initial state $\mathbf{x}_0 = \mathbf{x}(0)$, evaluated at time $t > 0$.

A vector $\mathbf{x} \in \mathbb{R}^n$ is an *equilibrium* point of the system (1) if $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Since any equilibrium point can be shifted to the origin $\mathbf{0}$ via a change of variables, without loss of generality, we may always assume that the equilibrium point of interest occurs at the origin.

Lyapunov theory is concerned with the behavior of the solution $\phi(t; \mathbf{x}_0)$ where the initial state $\mathbf{x}_0$ is not at the equilibrium $\mathbf{0}$ but is "close" to it.

Definition 1 (*Lyapunov stability*). The equilibrium point $\mathbf{0}$ of (1) is

 (i) *stable*, if for any $\epsilon > 0$, there exists $\delta = \delta(\epsilon)$ such that

$$\|\mathbf{x}_0\| < \delta \implies \|\phi(t; \mathbf{x}_0)\| < \epsilon, \quad \forall t > 0, \tag{2}$$

here $\| \cdot \|$ denotes any norm defined on $\mathbb{R}^n$;

 (ii) *unstable*, if it is not stable,

 (iii) *asymptotically stable*, if it is stable, and δ can be chosen such that

$$\|\mathbf{x}_0\| < \delta \implies \lim_{t \to \infty} \phi(t; \mathbf{x}_0) = \mathbf{0}; \tag{3}$$

 (iv) *globally asymptotically stable* if it is stable, and for all $\mathbf{x}_0 \in \mathbb{R}^n$,

$$\lim_{t \to \infty} \phi(t; \mathbf{x}_0) = \mathbf{0}. \tag{4}$$

Intuitively, the equilibrium point $\mathbf{0}$ is stable if all solutions starting near the origin (meaning that the initial conditions are in a neighborhood of the origin) remain near the origin for all time. The equilibrium point $\mathbf{0}$ is asymptotically stable if all solutions starting at nearby points not only stay nearby but also converge to the equilibrium point as time approaches infinity. And the equilibrium point $\mathbf{0}$ is globally asymptotically stable if it is asymptotically stable for all initial conditions $\mathbf{x}_0 \in \mathbb{R}^n$. The stability is an important property in practice, because it means arbitrarily small perturbations of the initial state about the equilibrium point $\mathbf{0}$ result in arbitrarily small perturbations in the corresponding solution trajectories of (1).

A sufficient condition for stability of the zero equilibrium is the existence of a Lyapunov function, as shown in the following theorem.

Theorem 2 ([31, Theorem 4.1]). *Let $D \subset \mathbb{R}^n$ be a domain containing the equilibrium point $\mathbf{0}$ of* (1). *If there exists a continuously differentiable function $V : D \rightarrow \mathbb{R}$ such that*

$$V(\mathbf{0}) = 0, \qquad V(\mathbf{x}) > 0 \quad in\ D \setminus \{\mathbf{0}\}, \tag{5}$$

$$\dot{V}(\mathbf{x}) := \frac{\partial V}{\partial \mathbf{x}} \cdot \mathbf{f}(\mathbf{x}) \leq 0 \quad in\ D, \tag{6}$$

then the origin is stable. Moreover, if

$$\dot{V}(\mathbf{x}) < 0 \quad in\ D \setminus \{\mathbf{0}\}, \tag{7}$$

then the origin is asymptotically stable.

A function $V(\mathbf{x})$ satisfying the conditions (5) and (6) in Theorem 2 is commonly known as a *Lyapunov function*. And we can verify globally asymptotic stability of system (1) by using Lyapunov functions stated as follows.

Theorem 3 ([31, Theorem 4.2]). *Let the origin be an equilibrium point for* (1). *If there exists a continuously differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that*

$$V(\mathbf{0}) = 0, \qquad V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0}, \tag{8}$$

$$\|\mathbf{x}\| \longrightarrow \infty \implies V(\mathbf{x}) \longrightarrow \infty, \tag{9}$$

$$\dot{V}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \neq \mathbf{0}, \tag{10}$$

then the origin is globally asymptotically stable.

Remark that a function satisfying condition (9) is said to be radially unbounded.

It is known that globally asymptotic stability is very desirable, but in many applications it is difficult to achieve. When the equilibrium point is asymptotically stable, we are interested in determining how far from the origin the trajectory can be and still converge to the origin as t approaches ∞. This gives rise to the following definition.

Definition 4 (region of attraction). The region of attraction (ROA) Ω of the equilibrium point $\mathbf{0}$ is defined as $\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid \lim_{t \to \infty} \phi(t; \mathbf{x}) = \mathbf{0}\}$.

The ROA of the equilibrium point $\mathbf{0}$ is a collection of all points $\mathbf{x}$ such that any trajectory starting at initial state $\mathbf{x}$ at time 0 will be attracted to the equilibrium point. In the literature, the terms "domain of attraction" and "attraction basin" are also used instead of "region of attraction".

Definition 5 (positively invariant set). A set $M \subset \mathbb{R}^n$ is called a positively invariant set of the system (1), if $\mathbf{x}_0 \in M$ implies $\phi(t; \mathbf{x}_0) \in M$ for all $t \geq 0$. Namely, if a solution belongs to a positively invariant set M at some time instant, then it belongs to M for all future time.

In general, finding the exact ROA analytically might be difficult or even impossible. Usually, Lyapunov functions are used to find underestimates of the ROA, that is, sets contained in the region of attraction, as stated in the following theorem.

Theorem 6 ([20, Theorem 10]). *Let $D \subset \mathbb{R}^n$ be a domain containing the equilibrium point $\mathbf{0}$ of* (1). *If there exists a continuously differentiable function $V : D \to \mathbb{R}$ satisfying* (5) *and* (7), *then any region $\Omega_c = \{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq c\}$ with $c \geq 0$ such that $\Omega_c \subseteq D$ is a positively invariant set contained in the ROA of the equilibrium $\mathbf{0}$.*

Theorem 6 describes an approach to compute estimates of the ROA through Lyapunov functions. More specifically, in the case of asymptotic stability, if $\Omega_c = \{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq c\}$ is bounded and contained in D, then every trajectory starting in Ω_c remains in Ω_c and approaches the origin as $t \to \infty$. Thus, Ω_c is an estimate of the ROA. Remark that when the origin is globally asymptotically stable, the region of attraction is the whole space $\mathbb{R}^n$.

3. Problem Reformulation

In this section, we will transform the problem of the asymptotic stability and ROA analysis of system (1) to a parametric program with LMI or BMI constraints. In the sequel, we suppose that system (1) is a polynomial dynamical system with $f_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ for $1 \leq i \leq n$, where $\mathbb{R}[\mathbf{x}]$ denotes the ring of real polynomials in the variables $\mathbf{x}$.

3.1. Asymptotic Stability. Firstly, we consider the asymptotic stability of system (1). As shown in Theorem 2, the existence of a Lyapunov function $V(\mathbf{x})$ which satisfies the conditions (5) and (7) is a certificate for asymptotical stability of the equilibrium point $\mathbf{0}$, and the problem of computing such a $V(\mathbf{x})$ can be transformed into the following problem:

$$
\begin{aligned}
\text{find} \quad & V(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] \\
\text{s.t.} \quad & V(\mathbf{0}) = 0, \\
& V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \in D \setminus \{\mathbf{0}\}, \\
& \dot{V}(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in D \setminus \{\mathbf{0}\}.
\end{aligned}
\tag{11}
$$

In general, D can be an arbitrary neighborhood of the equilibrium point $\mathbf{0}$. However, to simplify calculation, in practice, we can assume $D = \{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) \leq 0\}$, where $g(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is chosen to be $g(\mathbf{x}) = \sum_{i=1}^n x_i^2 - r^2$ for $1 \leq i \leq n$, with a given constant $r \in \mathbb{R}_+$, for instance, $r = 10^{-2}$.

Let us first predetermine a template of Lyapunov functions with the given degree d. We assume that

$$
V(\mathbf{x}) = \sum_\alpha c_\alpha \mathbf{x}^\alpha,
\tag{12}
$$

where $\mathbf{x}^\alpha = x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}$, $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ with $\sum_{i=1}^n \alpha_i \leq d$, and $c_\alpha \in \mathbb{R}$ are parameters. We can rewrite $V(\mathbf{x}) = \mathbf{c}^T \cdot T(\mathbf{x})$, where $T(\mathbf{x})$ is the (column) vector of all terms in $x_1, \ldots, x_n$ with total degree $\leq d$, and $\mathbf{c} \in \mathbb{R}^\nu$, with $\nu = \binom{n+d}{n}$, is the coefficient vector of $V(\mathbf{x})$. In the sequel, we write $V(\mathbf{x})$ as $V(\mathbf{x}, \mathbf{c})$ for clarity.

For a polynomial $\varphi(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, we say that $\varphi(\mathbf{x})$ is positive definite (resp., positive semidefinite), if $\varphi(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ (resp., $\varphi(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$). Observe that, if $\varphi(\mathbf{x})$ is a sum of squares (SOS), then $\varphi(\mathbf{x})$ is globally nonnegative. And, to ensure positive definiteness of $\varphi(\mathbf{x})$, we can use a polynomial $l(\mathbf{x})$ of the form:

$$
l(\mathbf{x}) = \sum_{i=1}^n \epsilon_i x_i^k,
\tag{13}
$$

where $\epsilon_i \in \mathbb{R}_+$ and k is assumed to be even. Clearly, the condition that $\varphi(\mathbf{x}) - l(\mathbf{x})$ is an SOS polynomial guarantees the positive definiteness of $\varphi(\mathbf{x})$. Therefore, to ensure positive definiteness of the second and third constraints in (11), we can use two polynomials $l_1(\mathbf{x})$, $l_2(\mathbf{x})$ of the form of (13). It is notable that SOS programming can be applied to determine the nonnegativity of a multivariate polynomial over a semialgebraic set. Consider the problem of verifying whether the implication

$$
\bigwedge_{i=1}^m \left(p_i(\mathbf{x}) \geq 0 \right) \implies q(\mathbf{x}) \geq 0
\tag{14}
$$

holds, where $p_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ for $1 \leq i \leq m$ and $q(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. According to Stengle's Positivstellensatz, Schmüdgen's Positivstellensatz, or Putinar's Positivstellensatz [32], if there exist SOS polynomials $\sigma_i \in \mathbb{R}[\mathbf{x}]$ for $i = 0, \ldots, m$, such that

$$
q(\mathbf{x}) = \sigma_0(\mathbf{x}) + \sum_{i=1}^m \sigma_i(\mathbf{x}) p_i(\mathbf{x}),
\tag{15}
$$

then the assertion (14) holds. Therefore, the existence of SOS representations provides a sufficient condition for determining the nonnegativity of $q(\mathbf{x})$ over $\{\mathbf{x} \in \mathbb{R}^n : \wedge_{i=1}^{m} p_i(\mathbf{x}) \geq 0\}$.

Based on the above observation, the problem (11) can be further transformed into the following SOS programming:

$$
\begin{aligned}
\text{find} \quad & \mathbf{c} \in \mathbb{R}^\nu \\
\text{s.t.} \quad & V(\mathbf{0}, \mathbf{c}) = 0, \\
& V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x}) = \rho_1(\mathbf{x}) - \sigma_1(\mathbf{x}) g(\mathbf{x}), \\
& -\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) - l_2(\mathbf{x}) = \rho_2(\mathbf{x}) - \sigma_2(\mathbf{x}) g(\mathbf{x}),
\end{aligned}
\tag{16}
$$

where $\sigma_j(\mathbf{x})$ and $\rho_j(\mathbf{x})$ are SOS in $\mathbb{R}[\mathbf{x}]$ for $j = 1, 2$. Moreover, the degree bound of those unknown SOS polynomials σ_j, ρ_j is exponential in n, $\deg(V)$, $\deg(\mathbf{f})$, and $\deg(g)$. In practice, to avoid high computational complexity, we simply set up a truncated SOS programming by fixing a *priori* (much smaller) degree bound $2e$, with $e \in \mathbb{Z}_+$, of the unknown SOS polynomials. Consequently, the existence of a solution $\mathbf{c}$ of (16) can guarantee the asymptotic stability of the given system.

Similarly, the problem of computing the Lyapunov function $V(\mathbf{x})$ for globally asymptotic stability of system (1), that satisfies the conditions in Theorem 3 can be rewritten as the following problem:

$$
\begin{aligned}
\text{find} \quad & V(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] \\
\text{s.t.} \quad & V(\mathbf{0}) = 0, \\
& V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0}, \\
& \|\mathbf{x}\| \longrightarrow \infty \Longrightarrow V(\mathbf{x}) \longrightarrow \infty, \\
& \dot{V}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \neq \mathbf{0}.
\end{aligned}
\tag{17}
$$

By introducing two polynomials $l_i(\mathbf{x})$, $i = 1, 2$ of the form (13), the condition, that $V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x})$ is SOS, guarantees the radially unboundedness of $V(\mathbf{x}, \mathbf{c})$. Consequently, the problem (17) can be further transformed into the following SOS programming:

$$
\begin{aligned}
\text{find} \quad & \mathbf{c} \in \mathbb{R}^\nu \\
\text{s.t.} \quad & V(\mathbf{0}, \mathbf{c}) = 0, \\
& V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x}) = \rho_1(\mathbf{x}), \\
& -\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) - l_2(\mathbf{x}) = \rho_2(\mathbf{x}),
\end{aligned}
\tag{18}
$$

where $\rho_j(\mathbf{x})$ are SOS polynomials in $\mathbb{R}[\mathbf{x}]$ for $j = 1, 2$. The decision variables are the coefficients of all the polynomials appearing in (18), such as $V(\mathbf{x}, \mathbf{c})$, $l_i(\mathbf{x})$, and $\rho_j(\mathbf{x})$.

3.2. Region of Attraction. Suppose that the equilibrium point $\mathbf{0}$ is asymptotically stable. In this section, we will consider how to find a large enough underestimate of the ROA. In the case where the equilibrium point $\mathbf{0}$ is globally asymptotically stable, the ROA becomes the whole space $\mathbb{R}^n$.

Our idea of computing the estimate of ROA is similar to that of the algorithm in [20, Section 4.2.2]. Suppose that D is a semialgebraic set

$$
\{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq 1\}
\tag{19}
$$

given by a Lyapunov function $V(\mathbf{x})$. In order to enlarge the computed positively invariant set contained in the ROA, we define a variable sized region

$$
P_\beta = \{\mathbf{x} \in \mathbb{R}^n \mid p(\mathbf{x}) \leq \beta\},
\tag{20}
$$

where $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a fixed and positive definite polynomial, and maximize β subject to the constraint $P_\beta \subseteq D$.

Fix a template of V of the form (12). The problem of finding an estimate P_β of the ROA can be transformed into the following polynomial optimization problem:

$$
\begin{aligned}
\max_{\mathbf{c} \in \mathbb{R}^\nu} \quad & \beta \\
\text{s.t.} \quad & V(\mathbf{0}, \mathbf{c}) = 0, \\
& V(\mathbf{x}, \mathbf{c}) > 0 \quad \forall \mathbf{x} \in D \setminus \{\mathbf{0}\}, \\
& \dot{V}(\mathbf{x}, \mathbf{c}) = \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in D \setminus \{\mathbf{0}\}, \\
& p(\mathbf{x}) \leq \beta \vDash V(\mathbf{x}, \mathbf{c}) \leq 1.
\end{aligned}
\tag{21}
$$

By introducing two polynomials $l_i(\mathbf{x})$, $i = 1, 2$ of the form (13) and based on SOS relaxation, the problem (21) can be further transformed into the following SOS programming:

$$
\begin{aligned}
\max_{\mathbf{c} \in \mathbb{R}^\nu} \quad & \beta \\
\text{s.t.} \quad & V(\mathbf{0}, \mathbf{c}) = 0, \\
& V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x}) = \rho_1(\mathbf{x}) - \sigma_1(\mathbf{x})(V(\mathbf{x}, \mathbf{c}) - 1), \\
& -\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) - l_2(\mathbf{x}) = \rho_2(\mathbf{x}) - \sigma_2(\mathbf{x})(V(\mathbf{x}, \mathbf{c}) - 1), \\
& 1 - V(\mathbf{x}, \mathbf{c}) = \rho_3(\mathbf{x}) - \sigma_3(\mathbf{x})(p(\mathbf{x}) - \beta),
\end{aligned}
\tag{22}
$$

where $\sigma_i(\mathbf{x})$ and $\rho_i(\mathbf{x})$ are SOS in $\mathbb{R}[\mathbf{x}]$ for $1 \leq i \leq 3$. In (22), the decision variables are β and the coefficients of all the polynomials appearing in (22), such as $V(\mathbf{x}, \mathbf{c})$, $\sigma_i(\mathbf{x})$, and $\rho_i(\mathbf{x})$. Since β and the coefficients of $V(\mathbf{x}, \mathbf{c})$ and $\sigma_i(\mathbf{x})$ are unknown, some nonlinear terms that are products of these coefficients, will occur in the second, third, and fourth constraints of (22), which yields a nonconvex bilinear matrix inequalities (BMI) problem. We will discuss in Section 4 how to handle the SOS programming (22) directly using the BMI solver or iterative method.

4. Exact Certificate of Sum of Squares Decomposition

According to Theorems 2, 3, and 6, the key of asymptotic stability analysis and ROA estimation lies in finding a real

function $V(\mathbf{x})$ and a constant β satisfying the desired conditions. We will present a symbolic-numeric hybrid method, based on SOS relaxation and rational vector recovery, to compute the exact polynomial $V(\mathbf{x})$ and constant β.

4.1. Approximate Solution from LMI or BMI Solver.
In this section, we will discuss how to handle the SOS programmings (16), (18), and (22) directly using the LMI and BMI solver or iterative method.

Using Gram matrix representation method [23] (also known as square matrix representation (SMR) [33]), a polynomial $\psi(\mathbf{x})$ of degree $2m$ is SOS if and only if there exists a positive semidefinite matrix Q such that

$$\psi(\mathbf{x}) = Z(\mathbf{x})^T Q Z(\mathbf{x}), \tag{23}$$

where $Z(\mathbf{x})$ is a monomial vector in $\mathbf{x}$ of degree m. Thus, the SOS programming (16) is equivalent to the following semidefinite program (SDP) problem:

$$\inf_{W^{[j]},Q^{[j]}} \sum_{j=1}^{2} \mathrm{Trace}\left(W^{[j]} + Q^{[j]}\right)$$

s.t. $V(\mathbf{0}) = 0,$

$$V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x}) = m_1(\mathbf{x})^T \cdot W^{[1]} \cdot m_1(\mathbf{x}) \tag{24}$$
$$\qquad - p_1(\mathbf{x})^T \cdot Q^{[1]} \cdot p_1(\mathbf{x}) g(\mathbf{x}),$$

$$-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) - l_2(\mathbf{x}) = m_2(\mathbf{x})^T \cdot W^{[2]} \cdot m_2(\mathbf{x})$$
$$\qquad - p_2(\mathbf{x})^T \cdot Q^{[2]} \cdot p_2(\mathbf{x}) g(\mathbf{x}),$$

where all the matrices $W^{[j]}, Q^{[j]}$ are symmetric and positive semidefinite, and $\sum_{j=1}^{2} \mathrm{Trace}(W^{[j]} + Q^{[j]})$ denotes the sum of traces of all these matrices, which acts as a dummy objective function commonly used in SDP for optimization problem with no objective function.

Similarly, the SOS programming (18) can be rewritten as the following SDP problem:

$$\inf_{W^{[1]},W^{[2]}} \mathrm{Trace}\left(W^{[1]}\right) + \mathrm{Trace}\left(W^{[2]}\right)$$

s.t. $V(\mathbf{0}) = 0,$

$$V(\mathbf{x}, \mathbf{c}) - l_1(\mathbf{x}) = m_1(\mathbf{x})^T \cdot W^{[1]} \cdot m_1(\mathbf{x}), \tag{25}$$

$$-\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) - l_2(\mathbf{x}) = m_2(\mathbf{x})^T \cdot W^{[2]} \cdot m_2(\mathbf{x}),$$

where matrices $W^{[1]}, W^{[2]}$ are symmetric and positive semidefinite.

Many Matlab packages of SDP solvers, such as SOS-TOOLS [24], YALMIP [25], and SeDuMi [26], are available to solve the problem (24) and (25) efficiently.

Now, let us consider the problem (22). The following example shows how to transform nonlinear parametric polynomial constraints into a BMI problem.

Example 7. To find a polynomial $\varphi(x)$ satisfying $\varphi(x) \geq 0 \wedge -x^2 + 1 \geq 0 \vDash 2x(d\varphi/dx) \geq 0$, it suffices to find $\varphi(x)$ such that

$$2x \frac{d\varphi}{dx} = \phi_0(x) + \phi_1(x)\left(1 - x^2\right) + \phi_2(x)\varphi(x), \tag{26}$$

where $\phi_0(x), \phi_1(x), \phi_2(x)$ are SOSes. Suppose that $\deg(\varphi) = 1$, $\deg(\phi_0) = 2$ and $\deg(\phi_1) = \deg(\phi_2) = 0$ and that $\varphi(x) = u_0 + u_1 x$, $\phi_1 = u_2$ and $\phi_2 = v_1$, with $u_0, u_1, u_2, v_1 \in \mathbb{R}$ parameters. From (1) we have

$$\phi_0(x) = u_2 x^2 + \left(2u_1 - u_1 v_1\right)x - u_2 - u_0 v_1, \tag{27}$$

whose square matrix representation (SMR) [33] is $\phi_0(x) = Z^T Q Z$, where

$$Q = \begin{bmatrix} -u_2 - u_0 v_1 & u_1 - \dfrac{1}{2}u_1 v_1 \\ u_1 - \dfrac{1}{2}u_1 v_1 & u_2 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 \\ x \end{bmatrix}. \tag{28}$$

Since all the $\phi_i(x)$ are SOSes, we have $u_2 \geq 0$, $v_1 \geq 0$ and $Q \succeq 0$. Therefore, the original constraint is translated into a BMI constraint:

$$\mathcal{B}(u_0, u_1, u_2, v_1) = u_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} + u_2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$+ v_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + u_0 v_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$+ u_1 v_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\dfrac{1}{2} \\ 0 & 0 & -\dfrac{1}{2} & 0 \end{bmatrix} \succeq 0. \tag{29}$$

Similar to Example 7, the SOS programming (22) can be transformed into a BMI problem of the form

$$\inf \quad -\beta$$

$$\text{s.t.} \quad \mathcal{B}(\mathbf{u}, \mathbf{v}) = A_0 + \sum_{i=1}^{m} u_i A_i + \sum_{j=1}^{k} v_j A_{m+j} \tag{30}$$
$$\qquad\qquad + \sum_{1 \leq i \leq m} \sum_{1 \leq j \leq k} u_i v_j B_{ij} \succeq 0,$$

where A_i, B_{ij} are constant symmetric matrices, $\mathbf{u} = (u_1, \ldots, u_m)$, $\mathbf{v} = (v_1, \ldots, v_k)$ are parameter coefficients of the SOS polynomials occurring in the original SOS programming (22).

Many methods can be used to solve the BMI problem (30) directly, such as interior-point constrained trust region methods [34] an augmented Lagrangian strategy [35]. PENBMI

solver [27] is a Matlab package to solve the general BMI problem, whose idea is based on a choice of penalty/barrier function Φ_p that penalizes the inequality constraints. This function satisfies a number of properties [35] such that, for any $p > 0$, we have

$$\mathscr{B}(\mathbf{u}, \mathbf{v}) \succeq 0 \Longleftrightarrow \Phi_p(\mathscr{B}(\mathbf{u}, \mathbf{v})) \succeq 0. \tag{31}$$

This means that, for any $p > 0$, the problem (30) is equivalent to the following augmented problem:

$$\begin{aligned} \inf \quad & -\beta \\ \text{s.t.} \quad & \Phi_p(\mathscr{B}(\mathbf{u}, \mathbf{v})) \succeq 0. \end{aligned} \tag{32}$$

The associated Lagrangian of (32) can be viewed as a (generalized) augmented Lagrangian of (30):

$$F(\mathbf{u}, \mathbf{v}, U, p) = -\beta + Tr\left(U\Phi_p(\mathscr{B}(\mathbf{u}, \mathbf{v}))\right), \tag{33}$$

where U is the Lagrangian multiplier associated with the inequality constraint. Remark that U is a real symmetric matrix of the same order as the matrix operator $\mathscr{B}$. For more details please refer to [35].

Alternatively, observing (30), $\mathscr{B}(\mathbf{u}, \mathbf{v})$ involves no crossing products like $u_i u_j$ and $v_i v_j$. Taking this special form into account, an iterative method can be applied by fixing $\mathbf{u}$ and $\mathbf{v}$ alternatively, which leads to a sequential convex LMI problem [20]. Remark that although the convergence of the iterative method cannot be guaranteed, this method is easier to implement than PENBMI solver and can yield a feasible solution $(\overline{\mathbf{u}}, \overline{\mathbf{v}})$ efficiently in practice.

4.2. Exact SOS Recovery. Since the SDP (LMI) or BMI solvers in Matlab are running in fixed precision, applying the techniques in Section 4.1 only yields numerical solutions of (16), (18), and (22). In the sequel, we will propose an improved algorithm based on a modified Newton refinement and rational vector recovery technique, to compute exact solutions of polynomial optimization problems with LMI or BMI constraints.

Without loss of generality, we can reduce the problems (16), (18), and (22) to the following problem:

$$\begin{aligned} \text{find} \quad & \mathbf{c} \in \mathbb{R}^{\nu} \\ \text{s.t.} \quad & V_1(\mathbf{x}, \mathbf{c}) = \mathbf{m}_1(\mathbf{x})^T \cdot W^{[1]} \cdot \mathbf{m}_1(\mathbf{x}), \\ & V_2(\mathbf{x}, \mathbf{c}) = \mathbf{m}_2(\mathbf{x})^T \cdot W^{[2]} \cdot \mathbf{m}_2(\mathbf{x}) \\ & \qquad + \left(\mathbf{m}_3(\mathbf{x})^T \cdot W^{[3]} \cdot \mathbf{m}_3(\mathbf{x})\right) \cdot V_3(\mathbf{x}, \mathbf{c}), \\ & W^{[i]} \succeq 0, \quad i = 1, 2, 3, \end{aligned} \tag{34}$$

where the coefficients of the polynomials $V_i(\mathbf{x}, \mathbf{c})$, $1 \leq i \leq 3$ are affine in $\mathbf{c}$. Note that (34) involves both LMI and BMI constraints. After solving the SDP system (34) by applying the techniques in Section 4.1, the numerical vector $\mathbf{c}$ and the

numerical positive semidefinite matrices $W^{[i]}$, $i = 1, 2, 3$ may not satisfy the conditions in (34) exactly, that is,

$$V_1(\mathbf{x}, \mathbf{c}) \approx \mathbf{m}_1(\mathbf{x})^T \cdot W^{[1]} \cdot \mathbf{m}_1(\mathbf{x}),$$

$$V_2(\mathbf{x}, \mathbf{c}) \approx \mathbf{m}_2(\mathbf{x})^T \cdot W^{[2]} \cdot \mathbf{m}_2(\mathbf{x}) \tag{35}$$

$$+ \left(\mathbf{m}_3(\mathbf{x})^T \cdot W^{[3]} \cdot \mathbf{m}_3(\mathbf{x})\right) \cdot V_3(\mathbf{x}, \mathbf{c}),$$

as illustrated by the following example.

Example 8. Consider the following nonlinear system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \begin{aligned} &-0.5x_1 + x_2 + 0.24999x_1^2 \\ &+ 0.083125x_1^3 + 0.0205295x_1^4 \\ &+ 0.0046875x_1^5 + 0.0015191x_1^6 \\ &-x_2 + x_1x_2 - 0.49991x_1^3 + 0.040947x_1^5 \end{aligned} \end{bmatrix}. \tag{36}$$

We want to find a certified estimate of the ROA. In the associated SOS programming (22) with BMI constraints, we suppose $p(x_1, x_2) = x_1^2 + x_2^2$. When $d = 2$, we obtain

$$V(x_1, x_2) = 3.112937368x_1^2 + 3.112937368x_2^2,$$

$$\beta = 0.32124. \tag{37}$$

However, $V(x_1, x_2)$ and β cannot satisfy the conditions in (21) *exactly*, because there exists a sample point $(15/32, 85/256)$ such that the third condition in (21) cannot be satisfied. Therefore,

$$\Omega_V = \left\{ (x_1, x_2) \in \mathbb{R}^2 \mid V(x_1, x_2) \leq 1 \right\} \tag{38}$$

is not an estimate of the ROA of this system.

In our former papers [36, 37], we applied Gauss-Newton iteration and rational vector recovery to obtain exact solutions that satisfy the constraints in (34), *exactly*. However, these techniques may fail in some cases, as shown in [38, Example 2]. The reason may lie in that we recover the vector $\mathbf{c}$ and the associated positive semidefinite matrices separately. Here, we will compute exact solutions of (34) by using a modified Newton refinement and rational vector recovery technique [38], which applies on the vector $\mathbf{c}$ and the associated positive semidefinite matrices simultaneously. The main ideal is as follows.

We first convert $W^{[3]}$ to a nearby rational positive semidefinite matrix $\widetilde{W}^{[3]}$ by nonnegative truncated $PLDL^TP^T$-decomposition. In practice, $W^{[3]}$ is numerical diagonal matrix; in other words, the off-diagonal entries are very tiny and the diagonal entries are nonnegative. Therefore, by setting the small entries of $W^{[3]}$ to zeros, we easily get the nearby rational positive semidefinite matrix $\widetilde{W}^{[3]}$. We then apply Gauss-Newton iteration to refine $\mathbf{c}$, $W^{[1]}$, and $W^{[2]}$ simultaneously with respect to a given tolerance τ.

Now, we will discuss how to recover from the refined $\mathbf{c}$, $W^{[1]}$, $W^{[2]}$, the rational vector $\widetilde{\mathbf{c}}$, and rational positive semidefinite matrices $\widetilde{W}^{[1]}$ and $\widetilde{W}^{[2]}$ that satisfy *exactly*

$$V_1(\mathbf{x}, \widetilde{\mathbf{c}}) - \mathbf{m}_1(\mathbf{x})^T \cdot \widetilde{W}^{[1]} \cdot \mathbf{m}_1(\mathbf{x}) = 0,$$

$$V_2(\mathbf{x}, \widetilde{\mathbf{c}}) - \mathbf{m}_2(\mathbf{x})^T \cdot \widetilde{W}^{[2]} \cdot \mathbf{m}_2(\mathbf{x}) - V_3(\mathbf{x}, \widetilde{\mathbf{c}}) \mathbf{m}_3(\mathbf{x})^T \quad (39)$$

$$\cdot \widetilde{W}^{[3]} \cdot \mathbf{m}_3(\mathbf{x}) = 0.$$

Since the equations in (39) are affine in entries of $\widetilde{\mathbf{c}}$ and $\widetilde{W}_1$, $\widetilde{W}_2$, one can define an affine linear hyperplane

$$\mathscr{L} = \Big\{ \mathbf{c}, W^{[1]}, W^{[2]} \mid V_1(\mathbf{x}, \mathbf{c}) - \mathbf{m}_1(\mathbf{x})^T \cdot W^{[1]} \cdot \mathbf{m}_1(\mathbf{x}) = 0,$$

$$V_2(\mathbf{x}, \mathbf{c}) - \mathbf{m}_2(\mathbf{x})^T \cdot W^{[2]} \cdot \mathbf{m}_2(\mathbf{x}) - V_3(\mathbf{x}, \mathbf{c}) \mathbf{m}_3(\mathbf{x})^T$$

$$\cdot \widetilde{W}^{[3]} \cdot \mathbf{m}_3(\mathbf{x}) = 0 \Big\}. \quad (40)$$

Note that the hyperplane (40) can be constructed from a linear system $A\mathbf{y} = \mathbf{b}$, where $\mathbf{y}$ consists of the entries of $\mathbf{c}$, $W^{[1]}$ and, $W^{[2]}$. If A has full row rank, such a hyperplane is guaranteed to exist. Then, for a given bound D of the common denominator, the rationalized SOS solutions of (34) can be computed by orthogonal projection if the matrices $W^{[1]}$, $W^{[2]}$ have full ranks with respect to τ, or by the rational vector recovery otherwise. Finally, we check whether the matrices $\widetilde{W}^{[i]}$ are positive semidefinite for $i = 1, 2$. If so, the rational vector $\widetilde{\mathbf{c}}$ and rational positive semidefinite matrices $\widetilde{W}^{[i]}$ $1 \leq i \leq 3$ can satisfy the conditions of problem (34) exactly. For more details the reader can refer to [38].

4.3. Algorithm. The results in Sections 4.1 and 4.2 yield an algorithm to find exact solutions to the problem (34).

Algorithm 9. Verified Parametric Optimization Solver.
   Input

   (i) A polynomial optimization problem of the form (34).

   (ii) $D \in \mathbb{Z}_{>0}$: the bound of the common denominator.

   (iii) $e \in \mathbb{Z}_{\geq 0}$: the degree bound $2e$ of the SOSes used to construct the SOS programming.

   (iv) $\tau \in \mathbb{R}_{>0}$: the given tolerance.

   Output

   (i) The verified solution $\widetilde{\mathbf{c}}$ of (34) with the $\widetilde{W}^{[i]}, 1 \leq i \leq 3$ positive semidefinite.

      (1) (Compute the numerical solutions) Apply LMI or BMI solver to compute numerical solutions of the associated polynomial optimization problem (34). If this problem has no feasible solutions, return "we can't find solutions of (34) with the given degree bound $2e$." Otherwise, obtain $\mathbf{c}$ and $W^{[i]} \gtrsim 0, 1 \leq i \leq 3$.

   (2) (Compute the verified solution $\widetilde{\mathbf{c}}$)

      (2.1) Convert $W^{[3]}$ to a nearby rational positive semidefinite matrix $\widetilde{W}^{[3]}$ by non-negative truncated $\mathrm{PLDL^T P^T}$-decomposition.
      (2.2) For the tolerance τ, apply the modified Newton iteration to refine $\mathbf{c}$ and $W^{[1]}, W^{[2]}$.
      (2.3) Determine the singularity of $W^{[1]}$ and $W^{[2]}$ with respect to τ. Then, for a given common denominator, the rational vector $\widetilde{\mathbf{c}}$, and rational matrices $\widetilde{W}^{[1]}$, $\widetilde{W}^{[2]}$ can be obtained by orthogonal projection if $W^{[1]}$, $W^{[2]}$ are of full rank, or by rational vector recovery method otherwise.
      (2.4) Check whether the matrices $\widetilde{W}^{[1]}$, $\widetilde{W}^{[2]}$ are positive semidefinite. If so, return $\widetilde{\mathbf{c}}$, and $\widetilde{W}^{[i]}, 1 \leq i \leq 3$. Otherwise, return "we can't find the solutions of (34) with the given degree bound".

5. Experiments

Let us present some examples of asymptotic stability and ROAs analysis of nonlinear systems.

Example 10 ([8, Example 1]). Consider a nonlinear continuous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -x_1^3 + 4x_2^3 - 6x_3 x_4 \\ -x_1 - x_2 + x_5^3 \\ x_1 x_4 - x_3 + x_4 x_6 \\ x_1 x_3 + x_3 x_6 - x_4^3 \\ -2x_2^3 - x_5 + x_6 \\ -3x_3 x_4 - x_5^3 - x_6 \end{bmatrix}. \quad (41)$$

Firstly, we will certify the locally asymptotic stability of this system. According to Theorem 2, we need to find a Lyapunov function $V(\mathbf{x})$, which satisfies all the conditions in Theorem 2. We can set up an associated SOS programming (16) with $g(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 - 0.0001$. When $d = 4$, we obtain a feasible solution of the associated SDP system. Here we just list one approximate polynomial

$$V(\mathbf{x}) = 0.81569x_1^2 + 0.18066x_2^2 + 1.6479x_3^2 + 2.1429x_4^2$$

$$+ 1.2996x_6^2 + \cdots + 0.67105x_4^4 + 0.52855x_5^4. \quad (42)$$

Let the tolerance $\tau = 10^{-2}$, and let the bound of the common denominator of the polynomial coefficients vector be 100. By use of the rational SOS recovery technique described in Section 4.2, we can obtain an exact Lyapunov function and

the corresponding SOS polynomials. Here we only list the Lyapunov function:

$$\widetilde{V}(\mathbf{x}) = \frac{41}{50}x_1^2 + \frac{9}{50}x_2^2 + \frac{33}{20}x_3^2 + \frac{107}{50}x_4^2$$
$$+ \frac{69}{100}x_5^2 + \frac{13}{10}x_6^2 + \cdots + \frac{67}{100}x_4^4 + \frac{53}{100}x_5^4. \tag{43}$$

Therefore, the locally asymptotic stability is certified.

Furthermore, we will consider the globally asymptotic stability. It suffices to find a Lyapunov function $V(\mathbf{x})$ with rational coefficients, which satisfies all the conditions in Theorem 3. By solving the associated SOS programming (18), we obtain

$$V(\mathbf{x}) = 0.78997x_1^2 + 0.19x_2^2 + 2.2721x_3^2$$
$$+ 1.4213x_6^2 + \cdots + 1.4213x_2^4 + 0.71066x_5^4. \tag{44}$$

By use of the rational SOS recovery technique described in Section 4.2, we then obtain

$$\widetilde{V}(\mathbf{x}) = \frac{79}{100}x_1^2 + \frac{19}{50}x_1x_2 + \frac{9}{100}x_1x_5 + \frac{19}{100}x_2^2 + \frac{227}{100}x_3^2$$
$$+ \frac{199}{100}x_4^2 + \frac{9}{100}x_2x_5 + \frac{71}{100}x_5^2$$
$$+ \frac{71}{50}x_6^2 + \frac{71}{50}x_2^4 + \frac{71}{100}x_5^4, \tag{45}$$

which exactly satisfies the conditions in Theorem 3. Therefore, the globally asymptotic stability of this system is certified.

Table 1 shows the performance of Algorithm 9 on another six examples, for globally asymptotic stability analysis of dynamical systems in the literature. All the computations have been performed on an Intel Core 2 Duo 2.0 GHz processor with 2 GB of memory. In Table 1 Examples 1–3 correspond to [9, Examples 2, 3, 9] and Examples 4–6 correspond to [39, Example 7], [6, Example 22], and [40, page 1341]. For all these examples, let the degree bound of the SOSes be 4, and set $\tau = 10^{-2}$, $D = 100$. Here n and *Num* denote the number of system variables and the number of decision variables in the LMI problem, respectively; $\deg(\widetilde{V})$ denotes the degree of $\widetilde{V}(\mathbf{x})$ obtained from Algorithm 9; *Time* is that for the entire computation run Algorithm 9 in seconds.

Example 11 ([41, Example A]). Consider a nonlinear continuous system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 \\ x_1 + (x_1^2 - 1)x_2 \end{bmatrix}. \tag{46}$$

Firstly, we will certify the locally asymptotic stability of this system. It suffices to find a Lyapunov function $V(x_1, x_2)$ with rational coefficients, which satisfies all the conditions in Theorem 2. We can set up an associated SOS programming (16) with $g(\mathbf{x}) = x_1^2 + x_2^2 - 0.0001$. When $d = 2$, we obtain

$$V(\mathbf{x}) = 1.4957x_1^2 - 0.7279x_1x_2 + 1.1418x_2^2. \tag{47}$$

Table 1: Algorithm performance on benchmarks (globally asymptotic stability).

Example	n	Num	$\deg(\widetilde{V})$	Time (s)
1	2	11	2	1.391
2	2	11	2	1.418
3	3	18	2	1.674
4	4	81	4	2.951
5	2	24	4	1.832
6	2	24	4	1.830

Table 2: Algorithm performance on benchmarks (ROA).

Example	n	Num	$\deg(\widetilde{V})$	$\widetilde{\beta}$	β	Time (s)
1	2	28	2	5415/9277	0.593	19.417
2	3	46	2	2650/9902	2.76	26.635
3	2	28	2	99/41	2.05	9.915

Let the tolerance $\tau = 10^{-2}$, and, the bound of the common denominator of the polynomial coefficients vector be 100. By use of the rational SOS recovery technique described in Section 4.2, we then obtain a Lyapunov function:

$$\widetilde{V}(\mathbf{x}) = -\frac{62}{85}x_1x_2 + \frac{127}{85}x_1^2 + \frac{97}{85}x_2^2. \tag{48}$$

Therefore, the locally asymptotic stability is certified.

We now construct Lyapunov functions to find certified estimates of the ROA. In the associated SOS programming (22) with BMI constraints, we suppose $p(x_1, x_2) = x_1^2 + x_2^2$. When $d = 2$, we obtain

$$V(x_1, x_2) = 0.6174455x_1^2 - 0.40292x_1x_2 + 0.43078x_2^2,$$
$$\beta = 1.3402225. \tag{49}$$

Let the tolerance $\tau = 10^{-5}$, and the bound of the common denominator of the polynomial coefficients vector be 10^5. By use of the rational SOS recovery technique described in Section 4.2, we obtain

$$\widetilde{V}(x_1, x_2) = \frac{6732}{10903}x_1^2 - \frac{4393}{10903}x_1x_2 + \frac{42271}{10903}x_2^2,$$
$$\widetilde{\beta} = \frac{6701}{5000}, \tag{50}$$

which exactly satisfy the conditions in Theorem 6. Therefore,

$$\Omega_{\widetilde{V}} = \left\{ (x_1, x_2) \in \mathbb{R}^2 \mid \widetilde{V}(x_1, x_2) \le 1 \right\} \tag{51}$$

is a certified estimate of the ROA of the given system.

Table 2 shows the performance of Algorithm 9 on another three examples, for computing verified estimates of ROAs of dynamical systems with the same fixed positive definite

polynomial $p(\mathbf{x})$ given in the literature. All the computations have been performed on an Intel Core 2 Duo 2.0 GHz processor with 2 GB of memory. In Table 2, Examples 1–3 correspond to [22, Examples 1, 2] and [20, page 75]. For all these examples, let the degree bound of the SOSes be 4, $\tau = 10^{-4}$ and $D = 10000$. Here n and *Num* denote the number of system variables and the number of decision variables in the BMI problem, respectively; $\deg(\widetilde{V})$ and $\widetilde{\beta}$ denote, respectively, the degree of $\widetilde{V}(\mathbf{x})$ and value of β obtained from Algorithm 9, whereas β is reported in the literature; *time* is that for the entire computation run Algorithm 9 in seconds.

6. Conclusion

In this paper, we present a symbolic-numeric method on asymptotic stability and ROA analysis of nonlinear dynamical systems. A numerical Lyapunov function and an estimate of ROA can be obtained by solving an (bilinear) SOS programming via BMI solver. Then a method based on modified Newton iteration and rational vector recovery techniques is deployed to obtain exact polynomial Lyapunov functions and verified estimates of ROAs with rational coefficients. Some experimental results are given to show the efficiency of our algorithm. For future work, we will consider the problem of stability region analysis of nonpolynomial systems by applying a rigorous polynomial approximate technique to compute an uncertain polynomial system, whose set of trajectories contains that of the given nonpolynomial system.

Acknowledgments

References

[1] A. Polański, "Lyapunov function construction by linear programming," *Transactions on Automatic Control*, vol. 42, no. 7, pp. 1013–1016, 1997.

[2] T. A. Johansen, "Computation of Lyapunov functions for smooth nonlinear systems using convex optimization," *Automatica*, vol. 36, no. 11, pp. 1617–1626, 2000.

[3] L. Grüne and F. Wirth, "Computing control Lyapunov functions via a Zubov type algorithm," in *Proceedings of the 39th IEEE Confernce on Decision and Control*, vol. 3, pp. 2129–2134, IEEE, December 2000.

[4] R. C. L. F. Oliveira and P. L. D. Peres, "LMI conditions for robust stability analysis based on polynomially parameter-dependent Lyapunov functions," *Systems & Control Letters*, vol. 55, no. 1, pp. 52–61, 2006.

[5] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, "LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems," *International Journal of Robust and Nonlinear Control*, vol. 15, no. 1, pp. 35–49, 2005.

[6] J. Liu, N. Zhan, and H. Zhao, "Automatically discovering relaxed Lyapunov functions for polynomial dynamical systems," *Mathematics in Computer Science*, vol. 6, no. 4, pp. 395–408, 2012.

[7] T. Nguyen, Y. Mori, T. Mori, and Y. Kuroe, "QE approach to common Lyapunov function problem," *Journal of Japan Society for Symbolic and Algebraic Computation*, vol. 10, no. 1, pp. 52–62, 2003.

[8] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 3482–3487, IEEE, December 2002.

[9] Z. She, B. Xia, R. Xiao, and Z. Zheng, "A semi-algebraic approach for asymptotic stability analysis," *Nonlinear Analysis: Hybrid Systems*, vol. 3, no. 4, pp. 588–596, 2009.

[10] B. Grosman and D. R. Lewin, "Lyapunov-based stability analysis automated by genetic programming," *Automatica*, vol. 45, no. 1, pp. 252–256, 2009.

[11] K. Forsman, "Construction of Lyapunov functions using Grobner bases," in *Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 798–799, IEEE, December 1991.

[12] Z. She, B. Xue, and Z. Zheng, "Algebraic analysis on asymptotic stability of continuous dynamical systems," in *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, pp. 313–320, ACM, 2011.

[13] S. Prajna, P. A. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *Transactions on Automatic Control*, vol. 49, no. 2, pp. 310–314, 2004.

[14] B. Xia, "DISCOVERER: a tool for solving semi-algebraic systems," *ACM Communications in Computer Algebra*, vol. 41, no. 3, pp. 102–103, 2007.

[15] P. Giesl, "Construction of a local and global Lyapunov function using radial basis functions," *IMA Journal of Applied Mathematics*, vol. 73, no. 5, pp. 782–802, 2008.

[16] H.-D. Chiang and J. S. Thorp, "Stability regions of nonlinear dynamical systems: a constructive methodology," *Transactions on Automatic Control*, vol. 34, no. 12, pp. 1229–1241, 1989.

[17] E. J. Davison and E. M. Kurak, "A computational method for determining quadratic Lyapunov functions for non-linear systems," vol. 7, no. 5, pp. 627–636, 1971.

[18] E. Cruck, R. Moitie, and N. Seube, "Estimation of basins of attraction for uncertain systems with affine and Lipschitz dynamics," *Dynamics and Control*, vol. 11, no. 3, pp. 211–227, 2001.

[19] R. Genesio, M. Tartaglia, and A. Vicino, "On the estimation of asymptotic stability regions: state of the art and new proposals," *Transactions on Automatic Control*, vol. 30, no. 8, pp. 747–755, 1985.

[20] Z. Jarvis-Wloszek, *Lyapunov based analysis and controller synthesis for polynomial systems using sum-of-squares optimization [Ph.D. thesis]*, University of California, Berkeley, Calif, USA, 2003.

[21] S. Prakash, J. Vanualailai, and T. Soma, "Obtaining approximate region of asymptotic stability by computer algebra: a case study," *The South Pacific Journal of Natural and Applied Sciences*, vol. 20, no. 1, pp. 56–61, 2002.

[22] T. Weehong and A. Packard, "Stability region analysis using sum of squares programming," in *Proceedings of the American Control Conference (ACC '06)*, pp. 2297–2302, IEEE, June 2006.

[23] P. A. Parrilo, "Semidefinite programming relaxations for semi-algebraic problems," *Mathematical Programming B*, vol. 96, no. 2, pp. 293–320, 2003.

[24] S. Prajna, A. Papachristodoulou, and P. Parrilo, 2002, SOS-TOOLS: Sum of squares optimization toolbox for MATLAB, http://www.cds.caltech.edu/sostools .

[25] J. Löfberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," in *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control System Design*, pp. 284–289, September 2004.

[26] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 625–653, 1999.

[27] M. Kocvara and M. Stingl, *PENBMI User's Guide (version 2. 0)*, http://www.penopt.com, 2005.

[28] E. Kaltofen, B. Li, Z. Yang, and L. Zhi, "Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars," in *Proceedings of the 21th International Symposium on Symbolic and Algebraic Computation (ISSAC '08)*, pp. 155–164, ACM, Hagenberg, Austria, July 2008.

[29] E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi, "Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients," *Journal of Symbolic Computation*, vol. 47, no. 1, pp. 1–15, 2012.

[30] A. Platzer and E. M. Clarke, "The image computation problem in hybrid systems model checking," in *Proceedings of the 10th International Conference on Hybrid Systems: Computation and Control (HSCC '07)*, vol. 4416, pp. 473–486, Springer, April 2007.

[31] H. Khalil, *Nonlinear Systems*, Prentice hall, Upper Saddle River, NJ, USA, 3rd edition, 2002.

[32] J. Bochnak, M. Coste, and M. Roy, *Real Algebraic Geometry*, Springer, Berlin, Germany, 1998.

[33] G. Chesi, "LMI techniques for optimization over polynomials in control: a survey," *Transactions on Automatic Control*, vol. 55, no. 11, pp. 2500–2510, 2010.

[34] F. Leibfritz and E. M. E. Mostafa, "An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 1048–1074, 2002.

[35] M. Kočvara and M. Stingl, "Pennon: a code for convex nonlinear and semidefinite programming," *Optimization Methods & Software*, vol. 18, no. 3, pp. 317–333, 2003.

[36] M. Wu and Z. Yang, "Generating invariants of hybrid systems via sums-of-squares of polynomials with rational coefficients," in *Proceedings of the International Workshop on Symbolic-Numeric Computation*, pp. 104–111, ACM Press, 2011.

[37] W. Lin, M. Wu, Z. Yang, and Z. Zeng, "Exact safety verification of hybrid systems using sums-of-squares representation," *Science China Information Sciences*, 15 pages, 2012.

[38] Z. Yang, M. Wu, and W. Lin, "Exact verification of hybrid systems based on bilinear SOS representation," 19 pages, 2012, http://arxiv.org/abs/1201.4219 .

[39] A. Papachristodoulou and S. Prajna, "A tutorial on sum of squares techniques for systems analysis," in *Proceedings of the American Control Conference (ACC '05)*, pp. 2686–2700, IEEE, June 2005.

[40] J. Löfberg, "Modeling and solving uncertain optimization problems in YALMIP," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, pp. 1337–1341, July 2008.

[41] U. Topcu, A. Packard, and P. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, vol. 44, no. 10, pp. 2669–2675, 2008.