Proceedings of the 2007 American Control Conference
Marriott Marquis Hotel at Times Square
New York City, USA, July 11-13, 2007

Stability Region Analysis Using Simulations and Sum-of-Squares Programming

Ufuk Topcu, Andrew Packard, Peter Seiler and Timothy Wheeler

Abstract— **The problem of computing bounds on the region-of-attraction for systems with polynomial vector fields is considered. Invariant sets of the region-of-attraction are characterized as sublevel sets of Lyapunov functions. Finite dimensional polynomial parameterizations for the Lyapunov functions are used. A methodology utilizing information from simulations to generate Lyapunov function candidates satisfying necessary conditions for bilinear constraints is proposed. The suitability of the Lyapunov function candidates are assessed solving linear sum-of-squares optimization problems. Qualified candidates are used to compute provably invariant subsets of the region-of-attraction and to initialize various bilinear search strategies for further optimization. We illustrate the method on several small examples drawn from the literature.**

I. INTRODUCTION

The region-of-attraction (ROA) of a locally asymptotically stable equilibrium point is an invariant set such that all trajectories emanating from a point in it converge to the equilibrium point. It is an important tool in controller verification because perturbations from the equilibrium point within the ROA are guaranteed to stay in it and converge to the equilibrium point. Computing the exact ROA for nonlinear dynamics is very hard if not impossible. Therefore, researchers have focused on determining invariant subsets of the actual ROA. Among all other methods those based on Lyapunov functions are dominant in the literature [1], [2], [3], [4], [5], [6], [7]. These methods compute a Lyapunov function as a local stability certificate and sublevel sets of this Lyapunov function provide invariant subsets of the ROA. Due to recent advances in polynomial optimization based on sum-of-squares (SOS) relaxations [8], it is possible to search for polynomial Lyapunov functions for systems with polynomial or rational dynamics [4], [7].

Linear SOS optimization problems, those having an affine dependence of the decision variables, can be formulated as convex semidefinite programming problems [8]. However, the SOS relaxations for the problem of computing invariant subsets of the ROA lead to bilinear matrix inequality (BMI) constraints, see section III. BMIs are nonconvex and bilinear optimization problems, those with BMI constraints, are shown to be NP-hard in general [9]. The state-of-the-art of the solvers for the bilinear optimization problems is far behind that for the linear ones. Recently PENBMI, a solver for bilinear optimization problems, is introduced [10] and used for computing subsets of the ROA [7]. It is a local optimizer and its behavior (the speed of convergence and the quality of the local optimal point) depends on the initial point from which the optimization starts.

On the other hand, simulating a nonlinear system of moderate size, except those governed by stiff differential equations, is computationally efficient. Therefore, extensive simulation is still a tool used in real applications. Although the information from simulations is inconclusive, i.e., cannot be used to find provably invariant subsets of the ROA, it provides insight into the system behavior. For example, if using Lyapunov arguments, a function certifies that a given set is in the ROA, then that function must be a Lyapunov function on any solution trajectories initiating in that set (i.e., decreasing, positive). Using a finite number of convergent trajectories, and a linear parametrization of the Lyapunov function V, those constraints become LP constraints, and the feasible polytope (in V-coefficient space) is an outer bound on the set of valid Lyapunov functions. It is intuitive that drawing samples from this set to seed the nonconvex BMI solver may improve the performance of the solver (speed of convergence, convergence to a better local optimum, etc.). In fact, if there are a large number of simulation trajectories, samples from the set often are suitable Lyapunov functions (without further optimization) themselves. The idea of using simulation to constrain the search for Lyapunov functions is the main point of the paper. We illustrate the effectiveness through a series of examples.

Effectively, we are relaxing the bilinear problem (using a very specific system theoretic interpretation of the bilinear problem) to a linear problem, and the linear problem's feasible set outerbounds the true feasible set. Sampling the feasible set of the relaxed problem gives candidate solutions and/or seeds for the original problem.

[11] proposes a general relaxation for bilinear problems based on replacing bilinear terms by new variables and non-convex equality constraints by convex inequality constraints. This relaxation increases the dimension of decision variable space, so that the true feasible is a low dimension manifold in the relaxed feasible space. There may be efficient manners to correctly "project" samples of the relaxed problem into appropriate samples for the original problem, but we do not pursue this here.

The paper is organized as follows: The characterization of the invariant subsets of the ROA is explained in section II. The problem is transformed to a bilinear SOS program in section III. Section IV is devoted to the introduction of the proposed methodology. Illustration of the methodology

This work was supported by AFOSR.
U. Topcu, A. Packard and T. Wheeler are with the Department of Mechanical Engineering, The University of California, Berkeley, 94720-1740, USA utopcu, pack, twheeler@jagger.me.berkeley.edu
P. Seiler is with the Honeywell Labs in Minneapolis, MN, 55403, USA peter.j.seiler@honeywell.com

in section V is followed by concluding remarks.

Notation: $\mathcal{R}^n$ denotes n-dimensional real vector space and $\mathcal{C}^1$ denotes the set of real valued continuously differentiable functions on $\mathcal{R}^n$. A function $f : \mathcal{R}^n \to \mathcal{R}$ is called positive semidefinite (definite) if $f(x) \geq 0$ ($f(x) > 0$) for all $x \in \mathcal{R}^n$ (nonzero $x \in \mathcal{R}^n$). For a symmetric matrix $Q \in \mathcal{R}^{n\times n}$, $Q \succeq 0$ means Q is positive semidefinite. For an n-dimensional vector, $x \succeq 0$ means that $x_k \geq 0$ for $k = 1, \cdots, n$. $\mathbb{R}[x]$ represents the set of polynomials in x with real coefficients. The subset $\Sigma[x] := \{\pi \in \mathbb{R}[x] : \pi = \pi_1^2 + \pi_2^2 + \cdots + \pi_m^2,\ \pi_1, \cdots, \pi_m \in \mathbb{R}[x]\}$ of $\mathbb{R}[x]$ is the set of SOS polynomials. For $\pi \in \mathbb{R}[x]$, deg(π) denotes the degree of π.

II. Characterization of invariant subsets of the ROA

Consider the autonomous nonlinear dynamical system

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

where $x(t) \in \mathcal{R}^n$ is the state vector and $f : \mathcal{R}^n \to \mathcal{R}^n$ is such that $f(0) = 0$, i.e., the origin is an equilibrium point of (1) and f is locally Lipschitz on $\mathcal{R}^n$. Let $\phi(x, t)$ denote the solution to (1) with the initial condition $\phi(x, 0) = x$. If the origin is asymptotically stable but not globally attractive, one often wants to know which trajectories converge to the origin as time approaches ∞. This gives rise to the following definition of the *region-of-attraction*:

Definition 2.1: The region-of-attraction Ω of the origin for the system (1) is $\Omega := \{x \in \mathcal{R}^n : \lim_{t\to\infty} \phi(x, t) = 0\}$.

The following lemma, which is a modification of a similar result in [12], provides a characterization of invariant subsets of the ROA.

Lemma 2.1: [7] Let γ be a positive scalar. If there exists a continuously differentiable function $V : \mathcal{R}^n \to \mathcal{R}$ such that

$$V(0) = 0 \text{ and } V(x) > 0 \text{ for all } x \neq 0 \tag{2}$$

$$\Omega_{V,\gamma} := \{x \in \mathcal{R}^n : V(x) \leq \gamma\} \text{ is bounded, and} \tag{3}$$

$$\Omega_{V,\gamma} \setminus \{0\} \subseteq \{x \in \mathcal{R}^n : \nabla V(x) f(x) < 0\}, \tag{4}$$

then for all $x \in \Omega_{V,\gamma}$, the solution of (1) exists, satisfies $\phi(x, t) \in \Omega_{V,\gamma}$ for all $t \geq 0$, and $\lim_{t\to\infty} \phi(x, t) = 0$, i.e., $\Omega_{V,\gamma}$ is an invariant subset of Ω.

In order to enlarge the computed invariant subset of the ROA, we define a variable sized region $\mathcal{E}_\beta := \{x \in \mathcal{R}^n : p(x) \leq \beta\}$, where $p \in \mathbb{R}[x]$ is a fixed, positive definite, convex polynomial, and maximize β while imposing the constraint $\mathcal{E}_\beta \subseteq \Omega_{V,\gamma}$ and constraints (2)-(4). This can be written as

$$\max_{V,\beta} \quad \beta \quad \text{such that (2), (3) and (4) hold, and } \mathcal{E}_\beta \subseteq \Omega_{V,\gamma}. \tag{5}$$

The usefulness of simulation in understanding the ROA for a given system is undeniable. Faced with the task of performing a stability analysis (eg., "for a given p, is $\mathcal{E}_\beta$ contained in the ROA?"), a pragmatic, fruitful and wise approach begins with a linearized analysis and at least a modest amount of simulation experiments. Certainly, just one divergent trajectory starting in $\mathcal{E}_\beta$ certifies that $\mathcal{E}_\beta \not\subset ROA$. Conversely, a large collection of only convergent trajectories hints to the likelihood that indeed $\mathcal{E}_\beta \subset ROA$. Suppose this latter condition is true, let $\{\mathbf{c}_i\}_{i=1}^{N_{conv}}$ be a finite set of convergent (to the origin) trajectories, with initial conditions in $\mathcal{E}_\beta$. In the course of simulation runs, divergent trajectories whose initial conditions are not in $\mathcal{E}_\beta$ may also get discovered, so label these $\{\mathbf{d}_j\}_{j=1}^{N_{div}}$.

With β and γ fixed, the set of Lyapunov functions which certify that $\mathcal{E}_\beta \subset \Omega$ are simply

$$\left\{V \in \mathcal{C}^1 : \mathcal{E}_\beta \subset \Omega_{V,\gamma}, \text{and } (2) - (4) \text{ hold}\right\}$$

Of course, this could be empty, but it must be a subset of the intersection of the three convex sets given below.

$$\begin{array}{l} \left\{V \in \mathcal{C}^1 : \nabla V(\mathbf{c}_i(t)) f(\mathbf{c}_i(t)) < 0, \quad \forall i, \forall t \geq 0\right\} \\ \left\{V \in \mathcal{C}^1 : 0 < V(\mathbf{c}_i(t)) \leq \gamma, \quad \forall i, \forall t \geq 0\right\} \\ \left\{V \in \mathcal{C}^1 : \gamma < V(\mathbf{d}_j(t)), \quad \forall j, \forall t \geq 0\right\} \end{array} \tag{6}$$

Informally, these conditions simply say that any V which verifies that $\mathcal{E}_\beta \subset ROA$ using (2)-(4) must itself take on values between 0 and γ and be decreasing on the trajectories starting in $\mathcal{E}_\beta$, and be greater than γ on divergent trajectories.

How is this used? Roughly, in the next section, we restrict V to be polynomial, and relax the conditions in Lemma 2.1 and (5) into a bilinear SOS problem (a bilinear SDP, a BMI). The decision variables are the coefficients of V and other multiplier variables, denoted M. The nonconvex problem is difficult to solve – even obtaining a feasible point can be challenging for most algorithms. However, the problem has more structure than a general BMI. If either V or M, the problem becomes a linear SDP. Moreover, the discussion above yields a convex outer bound on the set of feasible V. We will sample points from the outer bound set to obtain candidate V's, and then solve (5) for M, holding V fixed. This "qualifies" V as a certificate, and then further BMI optimization is executed, using this (V, M) pair as an initial seed.

III. Bilinear SOS problem

The problem in (5) is an infinite dimensional problem. In order to make it amenable to numerical optimization, we restrict V to be a polynomial in x of fixed degree. Furthermore, we use the well-known sufficient condition for polynomial positivity [8]: for any $\pi \in \mathbb{R}[x]$, if $\pi \in \Sigma[x]$, then π is positive semidefinite.

Now, let $l_1 \in \mathbb{R}[x]$ be a positive definite polynomial. Then, the constraint $V - l_1 \in \Sigma[x]$ is a sufficient condition for constraints (2) and (3). The constraints (4) and $\mathcal{E}_\beta \subseteq \Omega_{V,\gamma}$ are set containment constraints and the following lemma, a generalization of the S-procedure [13], gives a sufficient condition for these constraints.

Lemma 3.1: Given $g_0, g_1, \cdots, g_m \in \mathbb{R}[x]$, if there exist $s_1, \cdots, s_m \in \Sigma[x]$ such that $g_0 - \sum_{i=1}^m s_i g_i \in \Sigma[x]$, then

$$\{x \in \mathcal{R}^n : g_i(x) \geq 0,\ i = 1, \cdots, m\} \subseteq \{x \in \mathcal{R}^n : g_0(x) \geq 0\}.$$

Introducing another positive definite polynomial, $l_2 \in \mathbb{R}[x]$ and applying Lemma (3.1), we obtain sufficient conditions for the constraints (4) and $\mathcal{E}_\beta \subseteq \Omega_{V,\gamma}$. The problem in (5) leads to the following optimization problem

$$\beta_B^* := \max \beta \text{ subject to} \tag{7}$$
$$V \in \mathbb{R}[x], V(0) = 0, s_1, s_2, s_3 \in \Sigma[x], \beta > 0$$
$$V - l_1 \in \Sigma[x], \tag{8}$$
$$-\left((\beta - p)s_1 + (V - \gamma)\right) \in \Sigma[x], \tag{9}$$
$$-\left((\gamma - V)s_2 + \nabla V f s_3 + l_2\right) \in \Sigma[x], \tag{10}$$

where V, s_1, s_2, and s_3 are of fixed degree and V, s_1, s_2, s_3, and β are decision variables.

The optimization problem in (7)-(10) is a bilinear SOS problem because of the product terms βs_1 in (9) and $V s_2$ and $\nabla V f s_3$ in (10).

Note that for given β and V, (9) and (10) are affine in s_1, s_2, and s_3. Therefore, for a given V, problem (7)-(10) can be solved by bisection on β. Each bisection step requires solving a linear feasibility problem. We propose a method for easily generating qualified Lyapunov function candidates using information from simulations in the next section.

IV. Relaxation of the bilinear SOS problem using simulation data

A. Linear relaxation using simulation data

Let V be parameterized as

$$V(x) = \varphi(x)^T \alpha,$$

where $\alpha_\nu \in \mathcal{R}$ is the ν^{th} entry of the column vector $\alpha \in \mathcal{R}^{n_b}$, φ is n_b-dimensional vector of polynomials in x, and n_b is a positive integer. An outer bound for the intersection of sets in (6) described by finitely many constraints is the intersection of the sets

$$\begin{array}{l} \{V \ : \ V(\mathbf{c}_i(1)) \leq \gamma, \quad \forall i\} \\ \{V \ : \nabla V(\mathbf{c}_i(\tau))(\mathbf{c}_i(\tau)) f(\mathbf{c}_i(\tau)) \leq -l_2(\mathbf{c}_i(\tau)), \quad \forall i, \forall \tau\} \\ \{V \ : \ V(\mathbf{d}_j(1)) \geq \gamma + \epsilon, \quad \forall j\}, \end{array} \tag{11}$$

where τ is the index for a finite set of points on trajectories $\mathbf{c}_i$ and $\mathbf{d}_j$ and $\tau = 1$ corresponds to the initial point. Call this intersection $\mathcal{Y}_1$. Equivalently, $\mathcal{Y}_1$ is the set of $\alpha \in \mathcal{R}^{n_b}$ such that the following set of constraints holds

$$\begin{array}{l} \varphi(\mathbf{c}_i(1))^T \alpha \leq \gamma, \quad \forall i, \\ \left[\nabla V(\mathbf{c}_i(\tau)) f(\mathbf{c}_i(\tau))\right]^T \alpha \leq -l_2(\mathbf{c}_i(\tau)), \quad \forall i, \forall \tau \\ \varphi(\mathbf{d}_j(1))^T \alpha \geq \gamma + \epsilon, \quad \forall j \end{array} \tag{12}$$

and therefore $\mathcal{Y}_1$ is a polytope in $\mathcal{R}^{n_b}$. More compactly, let us write the inequalities in (12) as $\Phi^T \alpha \leq b$.

Another necessary condition for (10) can be obtained as follows. Define $\mathcal{L}(P) := (\nabla f(0))^T P + P(\nabla f(0))$, where $P = P^T \in \mathcal{R}^{n \times n}$ is positive definite and $x^T P x$ is the quadratic part of V. Then,

$$\mathcal{L}(P) \preceq -\epsilon I \tag{13}$$

for some small constant ϵ. Indeed, by (10) we have $-x^T \mathcal{L}(P) x - l_2(x) - \gamma s_2(x) + \text{higher degree terms} \geq 0$ for all $x \in \mathcal{R}^n$. Hence, for all $x \in \{x \in \mathcal{R}^n \ : \ x^T x < r\}$ for sufficiently small $r > 0$, $x^T \mathcal{L}(P) x \leq -l_2(x)\gamma - s_2(x)$ which leads to (13). Let $\mathcal{Y}_2 := \{\alpha \in \mathcal{R}^{n_b} \ : \ P = P^T \succeq 0 \text{ and (13) holds}\}$. It is well known that $\mathcal{Y}_2$ is convex [13].

Proposition 4.1: $\mathcal{Y}_3 := \{\alpha \in \mathcal{R}^{n_b} \ : \ (8) \text{ holds}\} \subset \mathcal{R}^{n_b}$ is convex.

Proof: Let $\alpha_1, \ \alpha_2 \in \mathcal{Y}_3$ and $\lambda \in [0, 1]$. Then, there exist symmetric matrices $Q_1, \ Q_2 \in \mathcal{R}^{n_b' \times n_b'}$ such that $Q_1, \ Q_2 \succeq 0$, $\varphi(x)^T \alpha_i - l_1(x) = \psi(x)^T Q_i \psi(x)$, $i = 1, 2$, where $\psi(x)$ is a n_b' vector of monomials such that for all $i \in \{1, \cdots, n_b\}$, $\varphi_i(x) = \psi_j(x)\psi_k(x)$ for some $j, \ k \in \{1, \cdots, n_b'\}$. Then, $\varphi(x)^T (\lambda\alpha_1 + (1-\lambda)\alpha_2) - l_1(x) = \lambda(\varphi(x)^T \alpha_1 - l_1(x)) + (1-\lambda)(\varphi(x)^T \alpha_2 - l_1(x)) = \lambda\psi(x)^T Q_1 \psi(x) + (1-\lambda)\psi(x)^T Q_2 \psi(x) = \psi(x)^T \left[\lambda Q_1 + (1-\lambda) Q_2\right] \psi(x)$. Since $Q_1 \succeq 0$ and $Q_2 \succeq 0$ and the set of positive semidefinite matrices is convex, $\lambda Q_1 + (1-\lambda) Q_2 \succeq 0$. Hence, $\varphi^T (\lambda\alpha_1 + (1-\lambda)\alpha_2) - l_1 \in \Sigma[x]$ and $\mathcal{Y}_3$ is convex. □

Since $\mathcal{Y}_1$, $\mathcal{Y}_2$ and $\mathcal{Y}_3$ are convex, $\mathcal{Y} := \cap_{i=1}^3 \mathcal{Y}_i$ is a convex subset of the space of the coefficients of V. (12) constitutes a set of necessary conditions for (10) and (13); thus

$$\begin{array}{r} \mathcal{Y} \supset \mathcal{B} := \{\alpha \in \mathcal{R}^{n_b} \ : \ \exists s_2, s_3 \in \Sigma[x] \text{ such that} \\ (8) \text{ and } (10) \text{ hold}\}. \end{array} \tag{14}$$

This inclusion is depicted for an illustrative case in Fig. (1). Since (10) is not jointly convex in V and the multipliers, $\mathcal{B}$, the shaded region in Fig. (1), may not a convex set and even may not be connected.

A point in $\mathcal{Y}$ can be computed as any feasible point for the constraints (8), (12) and (13) which is a semidefinite program. An arbitrary point in $\mathcal{Y}$ may or may not be in $\mathcal{B}$. However, if we generate points (approximately) uniformly distributed in $\mathcal{Y}$, it is likely that some of the points are in $\mathcal{B}$. Therefore we want to generate a collection $\mathcal{A} := \{\alpha^{(k)}\}_{k=0}^{N_V - 1}$ of N_V points distributed approximately uniformly in $\mathcal{Y}$. For this we require $\mathcal{Y}$ to be bounded. If $\mathcal{Y}$ determined by the constraints (8), (12) and (13) is not bounded, additional constraints, such as bounding the absolute value of α_ν, can be used to make $\mathcal{Y}$ bounded.

B. Hit-and-Run Random Point Generation Algorithm

The following random point generation algorithm is used to generate points in $\mathcal{Y}$ [12]:

Algorithm 1: Given a positive integer N_V and $\alpha^{(0)} \in \mathcal{Y}$, set $k = 0$

i. Generate a random vector $\xi^{(k)} = y^{(k)}/\|y^{(k)}\|$, where $y^{(k)} \sim \mathcal{N}(0, I_{n_b})$;
ii. Compute the extreme values of $\underline{t}^{(k)} \leq 0$ and $\overline{t}^{(k)} \geq 0$ such that $\alpha^{(k)} + \underline{t}^{(k)}\xi^{(k)}$ and $\alpha^{(k)} + \overline{t}^{(k)}\xi^{(k)}$ are on the boundary of $\mathcal{Y}$;
iii. Pick $w^{(k)}$ from a uniform distribution on $[0, 1]$;
iv. $\alpha^{(k+1)} = w^{(k)}(\alpha^{(k)} + \underline{t}^{(k)}\xi^{(k)}) + (1 - w^{(k)})(\alpha^{(k)} + \overline{t}^{(k)}\xi^{(k)})$
v. $k = k + 1$
vi. If $k = N_V$, return $\mathcal{A}$; else, go to (i).

Fig. (1) shows the initial point (big dot), an arbitrary point satisfying (8), (12) and (13), and the points generated by

Algorithm 1 (small dots) for the illustrative example. Since $\mathcal{Y}$ is a convex, compact subset of $\mathcal{R}^{n_b}$ and points are generated as convex combinations of points in $\mathcal{Y}$ (step (iv)), the algorithm generates N_V points in $\mathcal{Y}$. For sufficiently large N_V, points of $\mathcal{A}$ become approximately uniformly distributed in $\mathcal{Y}$ [14].

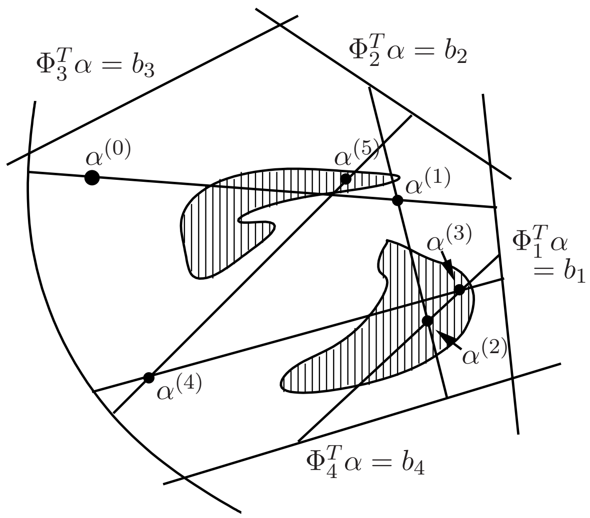

Fig. 1. Regions $\mathcal{Y}$ and $\mathcal{B}$ and points generated by the hit-and-run random point generation algorithm. Φ_j and b_j denote the j^{th} column of Φ and j^{th} row of b.

In step (ii), $\underline{t}^{(k)}$ and $\bar{t}^{(k)}$ are determined by

$$\begin{aligned}
\bar{t}^{(k)} &:= \min \left\{ \max_j \left\{ 0, \frac{b_j - \Phi_j^T\alpha^{(k)}}{\Phi_j^T\xi^{(k)}} \right\}, \bar{t}_{SOS}^{(k)}, \bar{t}_{lin}^{(k)} \right\}, \\
\underline{t}^{(k)} &:= \max \left\{ \min_j \left\{ 0, \frac{b_j - \Phi_j^T\alpha^{(k)}}{\Phi_j^T\xi^{(k)}} \right\}, \underline{t}_{SOS}^{(k)}, \underline{t}_{lin}^{(k)} \right\}, \\
\bar{t}_{SOS}^{(k)} &:= \max_{t\geq 0} \ t \text{ s.t. } \varphi(x)^T(\alpha^{(k)} + t\xi^{(k)}) \in \Sigma[x], \\
\bar{t}_{lin}^{(k)} &:= \max_{t\geq 0} \ t \text{ s.t. } L(P^{(k)} + t\Lambda^{(k)}) \preceq -\epsilon I,
\end{aligned}$$

where $P^{(k)} = {P^{(k)}}^T \in \mathcal{R}^{n\times n}$ and $\Lambda^{(k)} = {\Lambda^{(k)}}^T \in \mathcal{R}^{n\times n}$ are such that $x^T P^{(k)} x$ and $x^T\Lambda^{(k)}x$ are the quadratic parts of $\varphi(x)^T\alpha^{(k)}$ and $\varphi(x)^T\xi^{(k)}$, respectively. Similarly, $\underline{t}_{SOS}^{(k)}$ and $\underline{t}_{lin}^{(k)}$ are computed by imposing $t \leq 0$ in the optimization problems above instead of $t \geq 0$.

C. Algorithms

Since a feasible value of β is not known a priori, some iterative strategy to simulate and collect convergent and divergent trajectories is necessary. Our current strategy (used for the examples) to generate candidate Lyapunov functions is:

<u>*Algorithm 2*</u>: Given positive definite $p \in \mathbb{R}[x]$ with deg(p)=2 and constants β_{SIM}, N_{conv}, N_V, and $\beta_s \in (0, 1)$, set $\gamma = 1$, $N_{more} = N_{conv}$.

i. Integrate (1) from N_{more} initial conditions in the set $\{x \in \mathcal{R}^n \ : \ p(x) = \beta_{SIM}\}$;
ii. If there is any diverging trajectory, set β_{SIM} to the minimum of the minimum value of p along the diverging trajectories and $\beta_s\beta_{SIM}$, set $N_{more} = N_{more} -$ number of diverging trajectories, and go to (i); else, continue;
iii. Truncate the convergent trajectories such that their initial points are in $\mathcal{E}_{\beta_{SIM}}$;
iv. Find a feasible point for (8), (12), and (13). If (8), (12), and (13) are infeasible, set $\beta_{SIM} = \beta_s\beta_{SIM}$, and go to (iii); else, continue;
v. Generate N_V Lyapunov function candidates using *Algorithm 1*, and return β_{SIM} and Lyapunov function candidates.

We assess the Lyapunov function candidates solving the following optimization problems.

<u>*Problem 1*</u>: Given $V = \varphi^T\alpha$ and $l_2 \in \mathbb{R}[x]$ positive definite, solve

$$\begin{aligned}
\gamma_L^* := \quad & \max_{\gamma>0, s_1, s_2\in\Sigma[x]} \ \gamma \text{ subject to} \\
& -[(\gamma - V)s_2 + \nabla V f s_3 + l_2] \in \Sigma[x],
\end{aligned} \tag{15}$$

and if *Problem 1* is feasible,

<u>*Problem 2*</u>: Given $V = \varphi^T\alpha$, p, and γ_L^*, solve

$$\begin{aligned}
\beta_L^* := \quad & \max_{\beta>0, s_1\in\Sigma[x]} \ \beta \text{ subject to} \\
& -[(\beta - p)s_1 - (V - \gamma_L^*)] \in \Sigma[x].
\end{aligned} \tag{16}$$

Note that we allowed γ to be a variable in *Problem 1* in order to get largest γ satisfying (15) for fixed V. Those of the Lyapunov function candidates with larger β_L^* may provide qualified seeds for PENBMI. Results supporting this claim will be demonstrated in section V.

V. Examples

A. Van der Pol dynamics

The Van der Pol dynamics

$$\begin{aligned}
\dot{x}_1 &= -x_2, \\
\dot{x}_2 &= x_1 + (x_1^2 - 1)x_2
\end{aligned}$$

have an unstable limit cycle, the boundary of the ROA, and a stable equilibrium point at the origin. We used the shape factor $p(x) = x_1^2 + x_2^2$, Lyapunov function candidates of degrees 2, 4, and 6, and $N_{conv} = 200$ and found $N_{div} = 101$ diverging trajectories. We initialized *Algorithm 2* with $\beta_{SIM} = 3.00$ and used $\beta_s = 0.90$. We found feasible points for (8), (12), and (13) when we used $\beta_{SIM} = 1.44,\ 1.97,$ and 2.19 (in step (iv)) for the Lyapunov function candidates of degrees 2, 4, and 6, respectively. We picked $N_V = 50$ and solved *Problems 1* and *2* for these Lyapunov function candidates and seeded PENBMI with these solutions. Fig. (2) shows the histograms of β_L^* for these 50 Lyapunov function candidates and β_B^* obtained using PENBMI seeded with the solution of *Problems 1* and *2*. Seeded PENBMI runs took $3 - 8$ and $11 - 24$ seconds on a Pentium IV 2.8 GHz desktop PC for the Lyapunov function candidates of degrees 4 and 6, respectively, and all of them terminated successfully. In addition, we performed 10 unseeded PENBMI runs for degrees 4 and 6 and they took $50 - 250$ seconds with a success ratio 0.9 and $1000 - 2500$ seconds with success ratio 0.5 for the Lyapunov function candidates of degrees 4 and 6, respectively. Table (I) shows maximum value of β_L^* and β_B^* among those for the 50 Lyapunov function candidates

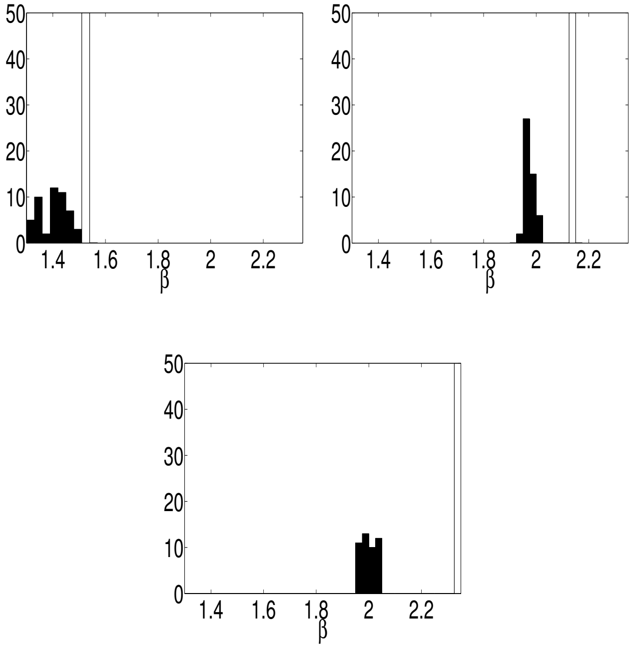

Fig. 2. Histograms of β_L^* (black bars) and β_B^* (white bars) from seeded PENBMI runs for V of degrees 2 (top left), 4 (top right), 6 (bottom).

TABLE I
MAXIMUM VALUE OF β_L^* AND β_B^* (FROM SEEDED PENBMI RUNS) AMONG 50 LYAPUNOV FUNCTIONS.

	deg(V)= 2	deg(V)= 4	deg(V)= 6
β_L^*	1.50	2.02	2.00
β_B^* (seeded)	1.52	2.14	2.34

and Fig. (3) shows the level sets of the Lyapunov functions corresponding to the value of β_B^* shown in Table (I). Note that the invariant subset of ROA certified by the Lyapunov function of degree 6 is visually indistinguishable from the limit cycle.

It is worth mentioning that the invariant subsets of ROA obtained using Lyapunov function candidates of degrees 4 and 6 are larger than those in [7]. A reason for this improvement is that we solve an optimization problem with more decision variables due to higher order multipliers in (7)-(10). This demonstrates the benefit of using higher degree multipliers when computationally tolerable.

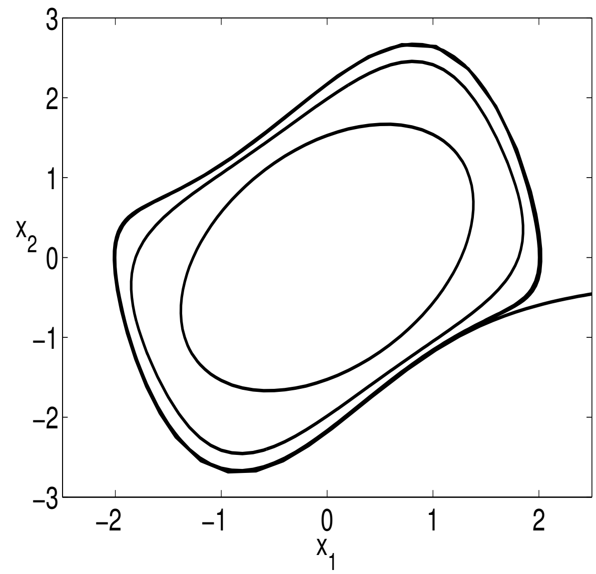

Fig. 3. Invariant subsets of the ROA and limit cycle (innermost curve for deg(V)=2, middle curve for deg(V)=4, and deg(V)=6 curve is indistinguishable from the outermost curve for the limit cycle).

B. Four examples from the literature

In this section we demonstrate results obtained using *Algorithm 2* for the systems

$$(E_1):\begin{cases}\dot{x}_1 = x_2,\\ \dot{x}_2 = -2x_1 - 3x_2 + x_1^2 x_2;\end{cases}$$

$$(E_2):\begin{cases}\dot{x}_1 = x_2,\\ \dot{x}_2 = (-2 + x_2^2 + x_2^4)x_1 - x_2 - x_1^5 + x_2^5;\end{cases}$$

$$(E_3):\begin{cases}\dot{x}_1 = x_2,\\ \dot{x}_2 = x_3,\\ \dot{x}_3 = -4x_1 - 3x_2 - 3x_3 + x_1^2 x_2 + x_1^2 x_3;\end{cases}$$

$$(E_4):\begin{cases}\dot{x}_1 = -x_2,\\ \dot{x}_2 = -x_3,\\ \dot{x}_3 = -0.915x_1 + (1 - 0.915x_1^2)x_2 - x_3,\end{cases}$$

where (E_1)-(E_3) are from [6] and (E_4) is from [5], [3]. We applied *Algorithm 2* with a quadratic Lyapunov function candidate, a shape factor $p(x) = x^T x$, and $N_V = 1$. Then, we used this quadratic V as the shape factor and applied *Algorithm 2* with a quartic Lyapunov function candidate and $N_V = 1$. Table (II) shows the ratio of the volume of the invariant subset of the ROA obtained using this procedure to that reported in [6] for (E_1), (E_2), and (E_3) along with the values of N_{conv} and N_{div}. We computed the empirical volume of a sublevel set of V by randomly sampling an hypercube containing the sublevel set. Fig. (4) shows $x_3 = 0$, $x_1 = 0$, and $x_2 = 0$ of the invariant subsets, computed by our method and reported in [5], of the ROA for (E_4).

VI. CONCLUSIONS

We considered the problem of computing the largest invariant subset of the region-of-attraction of a system with polynomial dynamics certified by a polynomial vector Lyapunov function of fixed degree. Similar to many local

TABLE II

VOLUME RATIOS, N_{conv}, AND N_{div} FOR $(E1)$, $(E2)$, AND $(E3)$.

	volume ratio	N_{conv}	N_{div}
(E1)	16.71/10.21	400	177
(E2)	0.99/0.85	400	181
(E3)	37.20/23.47	800	140

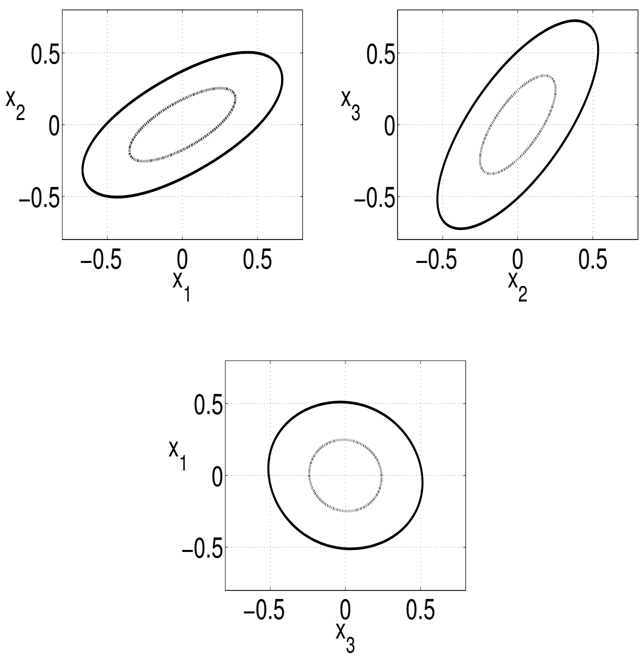

Fig. 4. $x_3 = 0$, $x_1 = 0$, and $x_2 = 0$ slices of the invariants subsets of the ROA for (E_4) (dot: [5], solid: *Algorithm 2*).

analysis problems, this is a nonconvex problem. Furthermore, its sum-of-squares relaxation leads to bilinear optimization problem. We proposed a method utilizing information from simulations for generating Lyapunov function candidates in a convex set containing the actual feasible set in the parameter space. We assessed these candidates solving linear sum-of-squares optimization problems, seeded a local bilinear search scheme with the solutions of these linear problems, and obtained promising results.

VII. ACKNOWLEDGMENTS

This work was sponsored by the Air Force Office of Scientific Research, USAF, under grant/contract number FA9550-05-1-0266. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFOSR or the U.S. Government.

The authors would like to thank Johan Lofberg for incorporating bilinear SOS parameterization into YALMIP [15] and developers of SOSTOOLS [16]. The authors would also like to thank Weehong Tan for valuable discussions.

REFERENCES

[1] H.-D. Chiang and J. S. Thorp, “Stability regions of nonlinear dynamical systems: A constructive methodology,” *IEEE Transaction on Automatic Control*, vol. 34, no. 12, pp. 1229–1241, 1989.

[2] R. Genesio, M. Tartaglia, and A. Vicino, “On the estimation of asymptotic stability regions: State of the art and new proposals,” *IEEE Transaction on Automatic Control*, vol. 30, no. 8, pp. 747–755, 1985.

[3] E. J. Davison and E. M. Kurak, “A computational method for determining quadratic Lyapunov functions for nonlinear systems,” *Automatica*, vol. 7, pp. 627–636, 1971.

[4] A. Papachristodoulou, “Scalable analysis of nonlinear systems using convex optimization,” Ph.D. Dissertation, Caltech, Jan. 2005.

[5] A. Vannelli and M. Vidyasagar, “Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems,” *Automatica*, vol. 21, no. 1, pp. 69–80, 1985.

[6] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, “LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems,” *Int. J. Robust Nonlinear Control*, vol. 15, pp. 35–49, 2005.

[7] W. Tan and A. Packard, “Stability region analysis using sum of squares programming,” in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 2297–2302.

[8] P. Parrilo, “Semidefinite programming relaxations for semialgebraic problems,” *Mathematical Programming Series B*, vol. 96, no. 2, pp. 293–320, 2003.

[9] O. Toker and H. Ozbay, “On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback,” in *Proc. American Control Conf.*, 1995, pp. 2525–2526.

[10] M. Kočvara and M. Stingl, “*PENBMI User's Guide (Version 2.0)*, available from `http://www.penopt.com`,” 2005.

[11] S. Boyd and L. Vandenberghe, “Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization,” in *Communications, Computation, Control and Signal Processing: A Tribute To Thomas Kailath*, A. Paulraj, V. Roychowdhuri, and C. Schaper, Eds. Kluwer, 1997, pp. 279–288.

[12] M. Vidyasagar, *Nonlinear Systems Analysis*, 2^{nd} ed. Prentice Hall, 1993.

[13] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia: SIAM, 1994.

[14] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. Springer, 2005.

[15] J. Lofberg, “YALMIP : A toolbox for modeling and optimization in MATLAB,” in *Proc. CACSD Conf.*, Taiwan, 2004, pp. 2525–2526.

[16] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, Available from `http://www.cds.caltech.edu/sostools` and `http://www.mit.edu/~parrilo/sostools`, 2004.