Research Article

Estimation of region of attraction for polynomial nonlinear systems: A numerical method

Larissa Khodadadi a, *, 1, Behzad Samadi b, 2, Hamid Khaloozadeh c, 3

a Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
b Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
c Department of Systems and Control, Industrial Control Center of Excellence, KNToos University of Technology, Tehran, Iran

1. Introduction

Region of attraction (ROA) of a locally asymptotically stable equilibrium point is an invariant set such that all trajectories starting inside this set converge to the equilibrium point. ROA is an important tool in the stability analysis of systems, because the size of the ROA shows that how much the initial points can be far away from the equilibrium point and trajectories can still converge. Finding the exact ROA is in general a very difficult problem. An alternative is to estimate the ROA by computing the largest possible invariant subset of the ROA. In many applications, finding the stable equilibrium points for a nonlinear system is not sufficient to analyze the behavior of system. Because, in practice, a stable equilibrium point with a very small neighborhood may not be so much different comparing to an unstable equilibrium point. Besides, the autonomous nonlinear dynamical system can have several equilibrium points or limit cycles such that the trajectories might converge to each of these points or cycles in case they are stable. Therefore, estimating the stability region of a nonlinear system is a topic of significant importance and has been studied extensively for example in [1–15]. Most computational methods aim to compute the boundary of an invariant set inside the ROA. These methods can be split into Lyapunov and non-Lyapunov methods. Lyapunov methods compute a Lyapunov function (LF) as a local stability certificate and sublevel sets of this LF provide invariant subsets of the ROA. With the recent advances in polynomial optimization based on sum of squares (SOS) relaxations, it is possible to search for polynomial LFs for systems with polynomial and/or rational dynamics. In the literature, the following forms of Lyapunov candidate functions have been employed to estimate the ROA for nonlinear systems: Rational LFs, Polyhedral LFs, Piecewise affine LFs, Polynomial LFs such as Single LFs and Composite polynomial LFs. Rational LFs that approach infinity on the boundary of the ROA are constructed iteratively in [1] motivated by Zubov’s work. References [1,2] presented methods based on the concept of a maximal LF, for estimating the ROA of an autonomous nonlinear system. In [3] a nonlinear quadratic system with a locally asymptotically stable equilibrium point in origin was considered. A method is then proposed to determine whether a given polytope belongs to the ROA of the equilibrium using polyhedral LFs. The authors of [4] proposed a method to construct piecewise affine LFs. They suggested a fan-like triangulation around the equilibrium. They showed that if a two dimensional system has an exponentially stable equilibrium, there will be a local triangulation scheme such that a piecewise affine LF exist for the system. The method

Keywords:
Region of attraction
Sum of squares programming
Lyapunov stability
Polynomial nonlinear systems
Van der Pol equation

1  Postal address: Hafez Ave, Tehran, Iran.
2  Postal address: Resaalat, Seyyed Khandan Bridge, Shariati Street, Tehran, Iran.
3  Postal address: Hesarak, Ashrafie Ashfani Blvd, Poonak Sq, Tehran, Iran.
proposed in [5] consists of estimating the ROA via the union of a continuous family of polynomial Lyapunov estimates rather than via one Lyapunov estimate. This method is formulated as a convex Linear Matrix Inequality (LMI) optimization by considering stability conditions for all candidate LFs at the same time [6,16]. In [7], stability analysis and controller synthesis of polynomial systems based on polynomial LFs were investigated. To search for LFs as well as the controllers and to prove local stability, the problem was formulated as iterative SOS optimization problems. In this work, the size of the ROA of the system was also estimated. The proposed method in [7] uses a variable-sized region defined by a shape factor to enlarge the estimation of the ROA. The goal is to find the largest sublevel set of a local LF that includes the largest possible shape factor region. Following [7], the authors of [8] proposed using bilinear SOS programming for enlarging a provable ROA of polynomial systems by polynomial LFs. Similar to [7], a polynomial was employed as a shape factor to enlarge the ROA estimation. For the same objective, the level sets of a polynomial LF of higher degree were employed because their level sets are richer than that of quadratic LFs. However, the number of optimization decision variables grows extremely fast as the degree of LF and the state dimension increases. In order to keep the number of decision variables low, using pointwise maximum or minimum of a family of polynomial functions was proposed. In [9], a methodology is proposed to generate LF candidates satisfying necessary conditions for bilinear constraints utilizing information from simulations. Qualified candidates were used to compute invariant subsets of the ROA and to initialize various bilinear search strategies for further optimization. In addition to Lyapunov-based methods, there are non-Lyapunov methods such as [10] that focus on topological properties of the ROA. For a survey of results, as well as an extensive set of examples, the reader is referred to [11].

In the last years, due to the importance of estimating the ROA in several fields such as clinical [17,18], economy [19], traffic [20], biological systems [21], chemical processes [22] etc., the ROA estimation has received considerable attention. This paper is motivated by the work in [7] that uses a shape factor to enlarge the estimation of the ROA. It will be shown that the choice of a proper shape factor is very important. However, no systematic method has been proposed to select or update the shape factor. Therefore, in this paper, we present a general algorithm for using a shape factor to enlarge the ROA estimation for nonlinear systems with polynomial vector fields. It will be shown that the proposed method is able to compute an estimation of the ROA of a benchmark problem that, to the best of our knowledge, is larger than the results obtained by existing methods.

This paper is organized as follows: Section 2 contains mathematical preliminaries. Problem statement and a Lyapunov-based method to estimate the ROA for nonlinear systems is explained in Section 3. Then, we propose the main result, an algorithm for selecting a shape factor to improve the estimation of the ROA in Section 4. Some numerical examples and simulation results have been shown in Section 5 to show the efficiency of the proposed algorithm. The paper closes with a conclusion and outlook in Section 6.

2. Mathematical preliminaries

Let the notation be as follows:

\( \mathbb{R}, \mathbb{Z}^+ \): the real number set and the positive integer set.
\( \mathbb{R}^n \): an \( n \)-dimensional vector space over the field of the real numbers.
\( \mathcal{P}_n \): the set of all polynomials in \( n \) variables.

Consider the autonomous nonlinear dynamical system

\[ \dot{x} = f(x) \]  

where \( x \in \mathbb{R}^n \) is the state vector and \( x(0) = x_0 \) is the initial state at \( t = 0 \) and \( f \in \mathcal{P}_n \) is a vector polynomial function of \( x \) with \( f(0) = 0 \). The origin is assumed to be locally asymptotically stable.

**Definition 1.** When the origin is asymptotically stable, the ROA of the origin is defined as

\[ \Omega := \{ x_0 \mid \lim_{t \to \infty} \varphi(t, x_0) = 0 \} \]  

where \( \varphi(t, x_0) \) is a solution of Eq. (1) that starts at initial state \( x_0 \).

**Definition 2.** A monomial \( m_a \) in \( n \) variables is a function defined as

\[ m_a := x_1^{a_1}x_2^{a_2}...x_n^{a_n} \]  

for \( a_i \in \mathbb{Z}^+ \). The degree of \( m_a \) is defined as \( \deg m_a := \sum_{i=1}^n a_i \).

**Definition 3.** A polynomial \( f \) in \( n \) variables is a finite linear combination of monomials,

\[ f := \sum_{a} c_a m_a = \sum_{a} c_a x^a \]  

with \( c_a \in \mathbb{R} \). The degree of \( f \) is defined as \( \deg f := \max \deg m_a \) (\( c_a \) is non-zero).

**Definition 4.** Define \( \mathcal{S}_n \) to be the set of SOS polynomials in \( n \) variables.

\[ \mathcal{S}_n := \{ p \in \mathcal{P}_n | p = \sum_{i=1}^k f_i^2, f_i \in \mathcal{P}_n, i = 1, ..., k \} \]  

Obviously if \( p \in \mathcal{S}_n \), then \( p(x) \geq 0 \) \( \forall x \in \mathbb{R}^n \). A polynomial, \( p \in \mathcal{S}_n \) if \( \exists \Omega \subseteq \mathbb{R}^n \) such that

\[ p(x) = z^T(x)Qz \]  

with \( z(x) \) a vector of suitable monomials [7].

**Definition 5.** Given \( \{ p_i \}_{i=0}^m \in \mathcal{P}_n \), generalized S-procedure states: if there exist \( \{ s_i \}_{i=1}^m \in \mathcal{S}_n \) such that \( p_0 - \sum_{i=1}^m s_i p_i \in \mathcal{S}_n \), then [23]

\[ \cap_{i=1}^m \{ x \in \mathbb{R}^n | p_i(x) \geq 0 \} \subseteq \{ x \in \mathbb{R}^n | p_0(x) \geq 0 \} \]  

3. Estimating the region of attraction

In general, exact computation of the ROA is a difficult task [9]. Hence, one should look for a numerical method to find a best possible estimation of the ROA. Since the Lyapunov technique is a powerful method in investigating the stability of nonlinear systems [24,25], in this section, a Lyapunov-based method is described to estimate the ROA by a LF sublevel set. The numerical algorithm in this section is based on a lemma from [26], that will be described in the following.

If for an open connected set \( S \) in \( \mathbb{R}^n \) containing \( 0 \), there exists a function \( V : \mathbb{R}^n \to \mathbb{R} \) such that \( V(0) = 0 \) and the following conditions hold

\[ V(0) > 0, \quad V(x) < 0, \quad \forall x \neq 0 \text{ in } S \]  

then every invariant set contained in \( S \) is also contained in the ROA of equilibrium point, but \( S \) itself need not be contained in ROA of \( 0 \). For finding such invariant sets, an easy way is to use so-called level sets of the (local) LF \( V \). Let \( c \) be a positive value, and consider the set

\[ M_V(c) = \{ x \in \mathbb{R}^n | V(x) \leq c \} \]  

Now, the connected level set \( M_V(c) \) containing \( 0 \) is a subset of the ROA of \( 0 \) [26].
The following corollary is a direct result of this fact.

**Corollary 1.** If there exists a continuously differentiable function 

\[ V : \mathbb{R}^n \rightarrow \mathbb{R} \] such that [8]

\[ V \text{ is positive definite} \] (10)

\[ \Omega := \{ x \in \mathbb{R}^n | V(x) \leq 1 \} \text{ is bounded} \] (11)

\[ \{ x \in \mathbb{R}^n | V(x) \leq 1 \} \setminus \{ 0 \} \subseteq \{ x \in \mathbb{R}^n | \frac{\partial V}{\partial x} < 0 \} \] (12)

Then for all \( x(0) \in \Omega \), the solution of Eq. (1) exists and

\[ \lim_{t \to +\infty} x(t) = 0. \]

As such, \( \Omega \) is a subset of the ROA for Eq. (1). The continuously differentiable function \( V \) is called a local LF.

In order to enlarge the \( \Omega \) (by choice of \( V \)), the author of [7] defines a variable sized region

\[ P_\beta := \{ x \in \mathbb{R}^n | p(x) \leq \beta \} \] (13)

and maximizes \( \beta \) while imposing the constraint \( P_\beta \subseteq \Omega \). Here \( \beta \) is a positive value and \( p(x) \) is a positive definite polynomial, chosen to reflect the relative importance of states, and it is called the shape factor. With the application of Corollary 1, the problem can be posed as the following optimization problem [7]:

Subject to: \( V(x) > 0 \) for all \( x \in \mathbb{R}^n \) and \( V(0) = 0 \) (14)

The set \( \{ x \in \mathbb{R}^n | V(x) \leq 1 \} \) is bounded (15)

\[ \{ x \in \mathbb{R}^n | p(x) \leq \beta \} \subseteq \{ x \in \mathbb{R}^n | V(x) \leq 1 \} \] (16)

\[ \{ x \in \mathbb{R}^n | V(x) \leq 1 \} /\{ 0 \} \subseteq \{ x \in \mathbb{R}^n | \frac{\partial V}{\partial x} < 0 \} \] (17)

Using the S-procedure and SOS programming, the following sufficient conditions can be obtained

\[ \max_{V \in \mathcal{S}_n, V(0) = 0, s_1, s_2 \in \Sigma_n} \beta \]

Subject to: \( V - I_2 \in \Sigma_n \) (18)

\[ -\left[ (\beta - p) s_1 + (V - 1) \right] \in \Sigma_n \] (19)

\[ -\left[ (1 - V) s_2 + \frac{\partial V}{\partial x} + I_2 \right] \in \Sigma_n \] (20)

where \( s_1 \) and \( s_2 \) are SOS polynomials and \( l_i(x) \) is a positive definite polynomial of the form [7]

\[ l_i(x) = \sum_{j=1}^{n} \epsilon_i x_j^2 \] (21)

for \( i = 1, 2 \) and \( \epsilon_i \) are positive numbers. Using SOS polynomials for solving the problem decreases generality of the problem. But instead possibility of converting the problem into SDP and solving it efficiently with numerical solvers is provided [27]. In this paper, we have used Yalmip [28] to convert SOS problems to SDP problems. Then, the Sedumi solver [29] has been used to solve the resulting SDP problems. Since there are products of decision variables in Eqs. (19,20), this is not a linear SOS programming.

Here we encounter SOS programs that are bilinear in the decision polynomials. Włoszek has proposed an iteration algorithm, holding one set of decision polynomials fixed and optimizing over the other set (which is an SDP), then switching over the sets which are optimized and holding the others fixed [7].

We illustrate this method with an example taken from [30].

**Example 3.1.** Consider the following Van der Pol system

\[ \begin{align*}
    \dot{x}_1 &= -x_2 \\
    \dot{x}_2 &= x_1 + 5(x_1^2 - 1)x_2
\end{align*} \] (22)

It has an unstable limit cycle and a stable equilibrium point at the origin. The problem of finding the ROA of the Van der Pol systems has been studied extensively, for example, in [1,7,8,9]. The ROA for this system is the region enclosed by its limit cycle.

In the following, numerical estimations of the ROA using different shape factors are presented.

Here we will try two different polynomials for \( p(x) \) and estimate the ROA. First we choose

\[ p(x) = x^2 \] (23)

For a sixth degree LF, the result is shown in Fig. 1. The second shape factor is a polynomial \( p(x) \) that aligns better with the shape of the ROA. Thus, the following \( p(x) \) is chosen

\[ p(x) = 0.2828x_1^2 - 0.1290x_1x_2 + 0.0359x_2^2 \] (24)

The result is shown in Fig. 2. It is seen that by using the shape factor in Eq. (24), the estimation is quite larger than the first one. As a result, the computed estimation can potentially be enhanced by choice of a proper shape factor. To the best of our knowledge, no algorithm has been proposed for choosing a proper

![Fig. 1. ROA estimation of Example 3.1. Boundary of true ROA (gray), level set of \( p(x) \) in the form of Eq. (23) (dotted) and estimated ROA (solid).](image1)

![Fig. 2. ROA estimation of Example 3.1. Boundary of true ROA (gray), level set of \( p(x) \) in the form of Eq. (24) (dotted) and estimated ROA (solid).](image2)
shape factor. To address this issue, in the next section, we will propose an algorithm.

4. Main result

In the previous section, it was shown that choosing the shape factor greatly affects the shape and the size of the ROA estimation. For second order systems, it might be possible to search for better choices of \( p(x) \) by looking at the phase portrait of the system. However, this is hardly possible for systems in higher dimensions where we are unable to visualize the ROA [7]. In [7,31], a completely uninspired polynomial as Eq. (23) was picked to enlarge the ROA estimation for different systems with ROAs of different shapes. Although when we do not have any information about the ROA, an uninspired choice of \( p(x) \) seems to be a good choice, a better option is to choose the shape factor based on the dynamics of each system, in order to compute a better estimation of the ROA. In the following, we present an algorithm for choosing the shape factor. Assuming that the origin is a locally asymptotically stable equilibrium point for the system in Eq. (1), the proposed algorithm to compute an estimation of the ROA is described in the following:

Algorithm 1

- **Initialization**: Compute the Jacobian matrix of \( f \) evaluated at \( x = 0 \):
  \[
  A = \frac{\partial f(x)}{\partial x} \bigg|_{x = 0} \tag{25}
  \]
  Then solve the equation
  \[
  A^T P + PA = -I \tag{26}
  \]
  for a positive definite matrix \( P \). Define
  \[
  V(x) = x^T P x \tag{27}
  \]
  \[
  p(x) = x^T f(x) \tag{28}
  \]
- **\( \gamma \) Step**: Hold \( V \) fixed and solve the following SOS programming for \( s_2(x) \)
  \[
  \begin{align*}
  & \text{Subject to:} \quad -\left[ (\gamma - V) s_2 + \frac{\partial V}{\partial x} f + l_2 \right] \in \sum_n \\
  \end{align*}
  \tag{29}
  \]
  where \( l_2 \) is defined in Eq. (21). There are products of decision variables, \( \gamma \) and \( s_2 \), in this step of the algorithm. Since \( \gamma \) is a positive scalar variable and \( s_2 \) is a polynomial, therefore we can use the bisection method described in [33] for solving the bilinear SOS problem.
- **\( \beta \) Step**: Hold \( V \) and \( p \) fixed and solve the following SOS programming for \( s_1(x) \)
  \[
  \begin{align*}
  & \text{Subject to:} \quad -[\beta - p] s_1 + (V - \gamma) \in \sum_n \\
  \end{align*}
  \tag{30}
  \]

Similar to \( \gamma \) Step, the bilinear problem in this step is also solved by bisection method.
- **\( V \) Step**: Hold \( s_1, s_2, \gamma, \beta \) and \( p \) fixed and compute \( V \) such that
  \[
  \begin{align*}
  & -\left[ \frac{\partial V}{\partial x} f + l_2 + s_2 (\gamma - V) \right] \in \sum_n \\
  \end{align*}
  \tag{31}
  \]
  \[
  -\left[ (V - \gamma) + s_1 (\beta - p) \right] \in \sum_n 
  \tag{32}
  \]

4.1. Updating the shape factor

In the previous section, it was shown that choosing the shape factor greatly affects the shape and the size of the ROA estimation. However, this is hardly possible for systems in higher dimensions where we are unable to visualize the ROA [7]. In [7,31], a completely uninspired polynomial as Eq. (23) was picked to enlarge the ROA estimation for different systems with ROAs of different shapes. Although when we do not have any information about the ROA, an uninspired choice of \( p(x) \) seems to be a good choice, a better option is to choose the shape factor based on the dynamics of each system, in order to compute a better estimation of the ROA. In the following, we present an algorithm for choosing the shape factor. Assuming that the origin is a locally asymptotically stable equilibrium point for the system in Eq. (1), the proposed algorithm to compute an estimation of the ROA is described in the following:

Algorithm 1

- **Initialization**: Compute the Jacobian matrix of \( f \) evaluated at \( x = 0 \):
  \[
  A = \frac{\partial f(x)}{\partial x} \bigg|_{x = 0} \tag{25}
  \]
  Then solve the equation
  \[
  A^T P + PA = -I \tag{26}
  \]
  for a positive definite matrix \( P \). Define
  \[
  V(x) = x^T P x \tag{27}
  \]
  \[
  p(x) = x^T f(x) \tag{28}
  \]
- **\( \gamma \) Step**: Hold \( V \) fixed and solve the following SOS programming for \( s_2(x) \)
  \[
  \begin{align*}
  & \text{Subject to:} \quad -\left[ (\gamma - V) s_2 + \frac{\partial V}{\partial x} f + l_2 \right] \in \sum_n \\
  \end{align*}
  \tag{29}
  \]
  where \( l_2 \) is defined in Eq. (21). There are products of decision variables, \( \gamma \) and \( s_2 \), in this step of the algorithm. Since \( \gamma \) is a positive scalar variable and \( s_2 \) is a polynomial, therefore we can use the bisection method described in [33] for solving the bilinear SOS problem.
- **\( \beta \) Step**: Hold \( V \) and \( p \) fixed and solve the following SOS programming for \( s_1(x) \)
  \[
  \begin{align*}
  & \text{Subject to:} \quad -[\beta - p] s_1 + (V - \gamma) \in \sum_n \\
  \end{align*}
  \tag{30}
  \]

Similar to \( \gamma \) Step, the bilinear problem in this step is also solved by bisection method.
- **\( V \) Step**: Hold \( s_1, s_2, \beta, \gamma \) and \( p \) fixed and compute \( V \) such that
  \[
  \begin{align*}
  & -\left[ \frac{\partial V}{\partial x} f + l_2 + s_2 (\gamma - V) \right] \in \sum_n \\
  \end{align*}
  \tag{31}
  \]
  \[
  -\left[ (V - \gamma) + s_1 (\beta - p) \right] \in \sum_n 
  \tag{32}
  \]

Remark 1. There are two major differences between the proposed algorithm and the V-s iteration in [9] that potentially lead to a larger ROA estimation. The first difference is that in the initialization step of the proposed algorithm, we set \( p \) as the LF for the linearized system (Eq. (28)), which is obtained from Eq. (26) instead of an uninspired shape factor of the form Eq. (23). Updating the shape factor in each iteration is the second major difference between the two algorithms. We update \( p \) by the quadratic term of the LF, which satisfies the local stability constraints. It is shown by various numerical examples that updating \( p \) in this way, greatly improves the ROA estimation. The resulting ROA estimation has been shown to be larger than the results of the previous works such as [7,9,31] that use a fixed shape factor to estimate the ROA.

Remark 2. \( V \) has to be positive definite and \( V(0) = 0 \), hence, it does not contain constant and linear terms. Also the degree of \( V \) should be even. The lack of a constant term in \( V(x) \) imposes some constraints on the selection of monomials in the SOS multipliers. For satisfying constraint in Eq. (19), we define the degree of \( s_1 \) such that \( \deg p + \deg s_1 \geq \deg V \). As we know, \( f(x) = 0 \) causes \( \partial V/\partial x \) not to contain any constant or linear terms. So for SOS conditions that require \( \partial V/\partial x f < 0 \) on the set \( |x|/V(x) \leq 1 \), the SOS multiplier associated with the term \( \gamma - V \) should not have a constant term. Therefore, in order to satisfy Eq. (20), we take the degree of \( s_2 \) such that \( \deg V + \deg s_2 \) be larger than the maximum degree of \( s_2, l_2 \) and \( \partial V/\partial x f \) [7,32].

Remark 3. For the initialization step of the algorithm, any \( p_q \) that is a solution of
  \[
  A^T p_q + P A = -Q \tag{34}
  \]
  with \( Q > 0 \) could have been chosen. In the following, we explain why \( Q = I \) is chosen. The dynamic equation of the system in Eq. (1) can be written as:
  \[
  \dot{x} = Ax + f_1(x) \tag{35}
  \]
  where \( Ax \) is the linearized part of the dynamics. Let
  \[
  f_1(x) = f(x) - Ax \tag{36}
  \]
  \[
  \lim_{|x| \rightarrow 0} \frac{|f_1(x)||}{|x|} = 0 \tag{37}
  \]

Define
  \[
  V(x) = x^T P_q x \tag{38}
  \]
  where \( P_q \) is the solution of Eq. (34). In [26], it is shown that:
  \[
  \dot{V}(x) \leq |x|[2\lambda_{max}(P_q)||f_1(x)|| - \lambda_{max}(Q)||x||] \tag{39}
  \]
  If \( r > 0 \) is chosen such that
  \[
  \frac{|f_1(x)||}{|x|} \leq \frac{\lambda_{max}(Q)}{2\lambda_{max}(P_q)} \quad \forall x \in B_r \tag{40}
  \]
  where \( B_r \) is the ball of radius \( r \), then \( \dot{V}(x) < 0 \) whenever \( x \in B_r \) and \( x \neq 0 \). As it is said in [26], every bounded level set of \( V(x) \) contained in \( B_r \) is also contained in the ROA.
Now Eq. (40) makes it clear that the larger the ratio \( \lambda_{\max}(Q)/\lambda_{\max}(P_2) \) is, the larger the possible choice of \( r \) becomes. Hence the “best” choice of \( Q \) is one that maximizes the ratio
\[
\mu(Q) = \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P_2)}
\]
where \( P_2 \) of course satisfies Eq. (34). It is shown in [26] that \( \mu(\cdot) \) is not affected by scaling \( Q \) and the identity matrix is the best choice for \( Q \) among positive definite matrices with \( \lambda_{\max}(Q) = 1 \). Therefore \( P \), the solution of Eq. (28) allows for the largest ball \( B_0 \) in which \( V(x) \) is negative.

Choosing \( Q = I \) provides only the largest ball so this choice can be conservative.

Remark 4. In order to guarantee the existence of a quadratic LF allowing to initialize the algorithm, we consider \( A \) in the Eq. (25) to be Hurwitz. Therefore the algorithm will always remain feasible if it is started from a feasible point. The feasible point is a LF such that there exist \( s_1 \) and \( s_2 \) that make the problem feasible [7].

Remark 5. In the proposed algorithm, the quadratic part of the computed \( V \) is employed to update \( P \). To justify this choice, the Van der Pol example in section 3 is explained with more details in the following. In the Van der Pol system in Eq. (23), the boundary of the ROA can be plotted using simulation data. The best ellipsoid that approximates the boundary of the ROA of this system is \( p(x) = 1 \) with the \( p \) in Eq. (24). Using this ellipsoid as the shape factor in the ROA estimation problem described by Eqs. (18)–(20), a local LF of degree six was obtained:
\[
V(x) = 0.1008x_1^2 + 0.08008x_2x_3 + 0.01761x_1x_2^2 - 0.01027x_1^2x_2^2 + 0.00254x_1x_2^3 - 0.0007x_1^3x_2 + 0.00013x_2^4 - 0.51883x_1^4 \\
- 0.1535x_2x_3 + 0.068856x_1x_2^2 - 0.00214x_1^2x_2 - 0.00028x_2^5 \\
+ 1.0696x_1^2 - 0.38864x_3 + 0.05370x_2^2
\]
(42)
The ellipsoid and the estimated ROA \( V(x) = 1 \) were shown in Fig. 2. When \( x \) is in the close neighborhood of the origin, \( V(x) \) can be approximated by its quadratic part:
\[
V_2(x) = 1.0696x_1^2 - 0.38864x_3x_2 + 0.05370x_2^2
\]
(43)
\[
V_2(x) \text{ can be rewritten as } \]
\[
V_2(x) = 3(0.3565x_1^2 - 0.1295x_1x_2 + 0.0179x_2^2)
\]
(44)
This shows that \( V_2(x) = 3 \) is an ellipsoid similar to the ellipsoid that was used as the shape factor: \( p(x) = 1 \) in Eq. (24). This similarity led us to the idea that the quadratic part of the local LF can be used as the shape factor for the next iteration of the algorithm. Although the inspiration came from approximating the exact ROA with an ellipsoid, the algorithm does not need to know the exact ROA and therefore simulation data is not used. It will be shown in the next section that this idea greatly improves the performance of the ROA estimation algorithm.

The proposed algorithm can potentially improve the ROA estimation of systems with an unbounded stability region by increasing the number of iterations. In previous work such as [8], it can happen that after a few iterations, even when the stability region is unbounded, the estimation does not improve because the shape factor is fixed and enlargement of the ROA estimation is limited with that fixed shape factor. Whereas in the proposed approach the shape factor can change its direction to cover a larger area inside the ROA in each iteration. At the initialization step of the proposed algorithm, there is still no computed local LF available. Therefore, the LF corresponding to the linearized dynamics of the system is used as the LF. This LF is quadratic so the initial shape factor is chosen equal to this LF. Since this choice is based on the dynamics of the system, it seems to be more efficient than an uninspired choice of \( p(x) \) as Eq. (23) for this example. The next section shows efficiency of the proposed algorithm.

5. Examples and simulation results

In this section, numerical examples are presented to illustrate the proposed method for enlarging the ROA estimation. These examples are second order systems with bounded and unbounded ROAs.

First, we present the systems which their exact stability boundaries are known and can easily be shown on phase portraits. Van der Pol equation in reverse time is a good example of this type of systems. We shall benchmark inner bounds on the ROA obtained with our methods against the exact stability boundaries. Then, we consider second order systems with unbounded ROAs. At the end of each example, we compare our method with previous work.

5.1. Systems with bounded ROAs

Example 5.1. In section 3 we discussed the Van der Pol equation in reverse time and estimated its ROA. Here we compute the ROA for that system with the proposed algorithm.
\[
\dot{x}_1 = -x_2 \\
\dot{x}_2 = x_1 + 5(x_1^2 - 1)x_2
\]
(45)
In initializing step of algorithm, matrices \( A \) and \( P \) are computed by using Eqs. (25) and (26) respectively:
\[
A = \begin{bmatrix} 0 & -1 \\ 1 & -5 \end{bmatrix}
\]
(46)
\[
P = \begin{bmatrix} 2.7 & -0.5 \\ -0.5 & 0.2 \end{bmatrix}
\]
(47)
we initialize the functions as follows:
\[
V(x) = 2.7x_1^2 - x_1x_2 + 0.2x_2^2
\]
(48)
\[
p(x) = 2.7x_1^2 - x_1x_2 + 0.2x_2^2
\]
(49)
Then, we hold \( V \) fixed and solve the Eq. (29). For solving the bilinear problem, we apply bisection method, described in [33]. To this aim, we take \( \gamma \in [0, 10] \) and define the bisection tolerance equal to \( 10^{-3} \). We also take \( l_2 = 10^{-6}(x_1^2 + x_2^2) \). In this step, we obtain \( \gamma = 1.1115 \) and
\[
s_2(x) = 0.5768x_1^2 + 0.28121x_1x_2 + 0.34161x_2^2 - 0.18304x_1x_2^2 \\
+ 0.13742x_2^2 + 0.53518x_1^2 - 0.55863x_1x_2 + 0.42883x_2^2
\]
(50)
In the next step, we hold \( V \) and \( P \) fixed and solve the Eq. (30). In this step, we also apply the bisection method by considering \( \beta \in [0, 10] \) and the bisection tolerance equal to \( 10^{-3} \). So, we obtain:
\[
s_1(x) = 0.29828x_1^2 - 0.06448x_1x_2 + 0.22045x_2^2 - 0.07636x_1^2 \\
+ 0.01596x_2^2 + 0.06618x_1^2 - 0.08892x_1x_2 - 0.07466x_2^2 + 0.92221
\]
(51)
and \( \beta = 1.1115 \). We take \( l_1 = 10^{-6}(x_1^2 + x_2^2) \).

In \( V \) step, we hold \( s_1 \) and \( s_2 \) and \( \beta \) and \( \gamma \) fixed. For achieving richer level sets of LFs, we search for a sixth degree polynomial LFs. By solving the Eq. (31)-(33), we have:
\[
V(x) = 0.20506x_1^2 - 0.00441x_1x_2 + 0.03987x_2^2 - 0.01754x_1^2 \\
+ 0.01075x_2^2 - 0.00343x_1 + 0.00083x_2^2 - 0.2089x_1^2 \\
- 0.1323x_1x_2 + 0.07401x_1x_2^2 + 0.01021x_1x_2 - 0.00863x_2^2
\]
We replace the quadratic terms of LF as shape factor:

\[ p(x) = 1.5876x_1^2 - 0.67042x_1x_2 + 0.17587x_2^2 \]  

These steps are then repeated to achieve an acceptable estimation. Here we experimentally set the number of iterations to 30. In this example, by using information of Lyapunov function from the 29th iteration, we choose the shape factor as:

\[ p(x) = 1.1046x_1^2 - 0.37796x_1x_2 + 0.042117x_2^2 \]  

Then we have:

\[
V(x) = 0.09720x_1^4 + 0.04312x_1^3x_2 - 0.00141x_1^2x_2^2 - 0.00890x_1^2 + 0.00056x_2^4 + 1.1112x_1^2 \\
+ 0.00232x_1^2 - 0.00025x_1x_2^2 - 0.52973x_1^2 - 0.10206x_1^2x_2 \\
+ 0.10739x_1^2x_2 - 0.01492x_1x_2^2 + 0.00056x_2^4 + 1.1112x_1^2 \\
- 0.37955x_1x_2 + 0.04222x_2^2 
\]  

Finally the level set of the \( V \) in Eq. (55) is the ROA estimation of this system. The results are shown in Fig. 3. Comparing with Fig. 1, the ROA estimation is enhanced saliently.

Example 5.2. The next example is taken from [5]:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -2x_1 - 3x_2 + x_1^2x_2 
\end{align*}
\]  

Chesi estimated the ROA for this system by using a union of quadratic LFs (Fig. 4). Using a sixth degree LF to estimate the ROA, the resulting ROA is shown in Fig. 5. This is again a significant improvement comparing to Fig. 4.

5.2. Systems with unbounded ROAs

Example 5.3. Consider the following system [34]:

\[
\begin{align*}
\dot{x}_1 &= -4x_1^3 + 6x_1^2 - 2x_1 \\
\dot{x}_2 &= -2x_2 
\end{align*}
\]  

Phase portrait of this system shows that the ROA of the equilibrium in the origin is not bounded.

Authors of [34] estimated the ROA of this system using a quartic LF. We also estimate the ROA of this system using a quartic LF. The result after 30 iterations is shown in Fig. 6. It is seen that the estimation is better than that of [34]. In this case, the ROA estimation can be improved by increasing the number of iterations without increasing the degree of the candidate LF. The result for 150 iterations is shown in Fig. 7.
The origin is a locally asymptotically stable equilibrium for this system. The ROA estimation computed in [3] and the result from the proposed algorithm are shown in Fig. 8.

Example 5.4. Consider the following second-order quadratic system [3]:

\[
\begin{align*}
    x_1 &= -50x_1 - 16x_2 + 13.8x_1x_2 \\
    x_2 &= 13x_1 - 9x_2 + 5.5x_1x_2
\end{align*}
\] (58)

The origin is a locally asymptotically stable equilibrium for this system. The ROA estimation computed in [3] and the result from the proposed algorithm are shown in Fig. 8.

Remark 6. Although the proposed algorithm improves the estimated ROA for the both of systems with bounded and unbounded ROAs, the estimation is bounded. Also for globally stable systems we cannot conclude that the ROA is the whole state space.

Remark 7. The main advantage of studying multivariate polynomial systems is that, polynomial systems can be used to approximate many nonlinear systems. Moreover, several methods exist for approximating the non-polynomial systems with polynomial ones. On the other hand, there are various available tools for stability analysis and design. For example, some powerful and promising relaxation techniques such as SOS programming are suitable for polynomial systems with polynomial LFs. Also, some softwares such as SOSTOOLS [35] and Gloptipoly [36] make the analysis of polynomial systems straightforward. All of these provide our motivation for investigating systems with polynomial descriptions.

6. Conclusion

In this paper, we proposed a method for estimating the ROA of nonlinear systems with polynomial vector fields. The proposed method contains an algorithm for updating the shape factor. The results of this method was demonstrated for a Van der Pol system in reverse time and also other systems with bounded and unbounded ROAs. These numerical examples showed the practical importance of the proposed method.

Appendix A

In this section, we explain bisection method for solving the existing non-convex optimization problem. The interval \([\gamma_{\text{lower}}, \gamma_{\text{upper}}]\) is guaranteed to contain \(\gamma^*\).

Given \(\gamma_{\text{lower}} \leq \gamma^*, \gamma_{\text{upper}} \geq \gamma^*\) and bisection tolerance > 0

While \((\gamma_{\text{upper}} - \gamma_{\text{lower}}) \geq\) bisection tolerance

1. \(\gamma_{\text{try}} = (\gamma_{\text{lower}} + \gamma_{\text{upper}})/2\)
2. Solve the convex feasibility problem in Eq. (1)
3. If the problem is feasible, \(\gamma_{\text{lower}} = \gamma_{\text{try}}\) else \(\gamma_{\text{upper}} = \gamma_{\text{try}}\)
4. End.

References

[34] Ratschan S, She Z. Providing a basin of attraction to a target region of polynomial systems by computation of lyapunov-like functions. SIAM Journal on Control and Optimization 2010;48:4377–94.