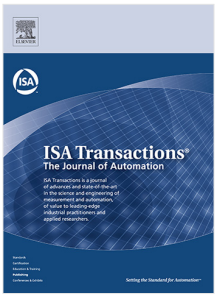
Journal Pre-proof



Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition

Dongyang Li, Dmitry Ignatyev, Antonios Tsourdos, Zhongyuan Wang

PII: S0019-0578(22)00607-3
DOI: https://doi.org/10.1016/j.isatra.2022.11.015
Reference: ISATRA 4867

To appear in: *ISA Transactions*

Received date : 17 December 2020
Revised date : 18 November 2022
Accepted date : 18 November 2022

Please cite this article as: D. Li, D. Ignatyev, A. Tsourdos et al., Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition. *ISA Transactions* (2022), doi: https://doi.org/10.1016/j.isatra.2022.11.015.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2022 Published by Elsevier Ltd on behalf of ISA.

ISA Transactions, Volume 136, May 2023, pp. 308-322
DOI:10.1016/j.isatra.2022.11.015

Journal Pre-proof

Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition

Dongyang Li[a,b,*,1], Dmitry Ignatyev[b,2], Antonios Tsourdos[b,2], Zhongyuan Wang[a,1]

[a] *School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*

[b] *School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 OAL, UK*

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Cranfield University; the China Scholarship Council under Grant No. 201906840024; and the Fundamental Research Funds for the Central Universities, China, No. 3091011401.

* Corresponding author. Tel.: +86 15651776672

E-mail address: dongylee@njust.edu.cn

[1] Postal address: School of Energy and Power Engineering, Nanjing University of Science and Technology, Xuanwu District, Nanjing 210094, China.

[2] Postal address: School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 OAL, UK.

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0). The final published version (version of record) is available online at DOI:10.1016/j.isatra.2022.11.015. Please refer to any applicable publisher terms of use.

Journal Pre-proof

Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition

Abstract

Sum-of-squares programming is widely used for region of attraction (ROA) estimations of asymptotically stable equilibrium points of nonlinear polynomial systems. However, existing methods yield conservative results, especially for non-symmetric and unbounded regions. In this study, a cost-effective approach for ROA estimation is proposed based on the Lyapunov theory and shape functions. In contrast to existing methods, the proposed method iteratively places the center of a shifted shape function (SSF) close to the boundary of the acquired invariant subset. The set of obtained SSFs yields robust ROA subsets, and R-composition is employed to express these independent sets as a single but richer-shaped level set. Several benchmark examples show that the proposed method significantly improves ROA estimations, especially for non-symmetric or unbounded ROA without a significant computational burden.

Keywords

Non-symmetric and unbounded region of attraction, Shape function, Polynomial nonlinear system, Sum of squares programming, Lyapunov stability

1 Introduction

Many real-world systems are governed by nonlinear equations [1]. Stability analysis of a nonlinear system in its general form using analytical techniques is difficult, if not impossible. Therefore, there is significant research interest in numerical methods that facilitate stability analysis. As opposed to linear time-invariant systems that allow for global analysis of stability, most nonlinear systems are stable only in a specific region around an equilibrium point (EP) [1]. For practical applications, it is important to determine the associated invariant set [2,3]. This invariant set, the so-called region of attraction (ROA) of the relevant EP, is an important

Journal Pre-proof

metric for system stability and robustness, and specifies the extent to which the initial states can be disturbed away from the expected steady states. Moreover, an actual dynamic system can have multiple stable EPs (or limit cycles) [2]. Therefore, the operative range of the system must be contained in the ROA of the expected EP. However, identification of the exact ROA is challenging both numerically and analytically [3]. Over the last few years, several studies have investigated techniques for obtaining a closer approximation of the exact ROA for different applications, such as aerospace, robotics, medical, chemical processes, traffic, and biological systems [2–24].

The most popular methods for ROA estimation are based on the Lyapunov function (LF). These facilitate the identification of the appropriate conditions for the stability of equilibria. Although the construction of LFs has been extensively studied [25], this is not a straightforward task and might not provide a solution for general cases. Many computational construction methods have been developed, together with various forms of relaxation, to develop a generalized approach for LF construction. A well-developed approach is the sum-of-squares (SOS) technique. The condition for a nonnegative multivariable polynomial is relaxed using an SOS polynomial [7]. For example, $s(x)$ is an SOS polynomial if there exist polynomials $\{f_i(x)\}_{i=1}^m \in \boldsymbol{R}(x)$ such that $s(x) = \sum_{i=1}^m f_i^2(x)$, where $\boldsymbol{R}[x]$ represents the set of polynomials in $x \in \mathbb{R}^n$ with real coefficients where m and n are positive integers [7]. The relaxed SOS constraint is more stringent but improves the tractability of both the construction of the LF and the computation of its invariant sublevel set, which is useful for ROA estimation. Detailed discussions on SOS techniques can be found in the literature [7,8,20,26].

SOS problems with constraints have been addressed via SOS programming. Freely available MATLAB toolboxes, such as SOSTOOLs [27] and SOSOPTs [28], transform SOS constraints into semidefinite problems (SDPs), which are then solved using SDP solvers such as SeDuMi [29]. For example, in SOS programming, the verification of a polynomial $s(x)$ as an SOS polynomial is equivalent to confirming the existence of a positive semidefinite matrix Q such that

$$s(x) = Z^T(x)QZ(x) \in \Sigma_n$$

where $Z(x)$ is an appropriately chosen vector of monomials in $x \in \mathbb{R}^n$ and Σ_n denotes the set of SOS polynomials in $x \in \mathbb{R}^n$. This is a basic feasibility problem in SOS programming, and another common class is the optimization problem for a linear objective function [26]. These problems are formulated using linear matrix inequalities (LMIs) and solved using LMI solvers. However, the formulation of some practical problems might result in bilinear matrix inequalities that cannot be easily solved using LMI solvers, which occurs during the process of identifying the largest possible invariant subset of the ROA [2,5]. In the ROA problem, the decision variables to be optimized are coupled with auxiliary SOS multipliers. One solution is to use bilinear solvers (such as PENMBI in YALMIP [30]). Although they facilitate direct treatment of bilinear problems, they are less developed than linear solvers, and convergence to a global optimum cannot be guaranteed [5,8,9,31,32]. The other approach is to convert it into a two-way iterative search between LFs and SOS multipliers using the additional structure of the ROA estimation problem rather than a general bilinear problem. It then becomes affine in SOS multipliers if LF is fixed and vice versa [27,33]. A two-way search can then be conveniently solved using widely used linear solvers.

Two-way iteration procedures have been investigated in many applications to improve these procedures. For example, an interior expanding algorithm [34] was proposed to enlarge ROA estimation using a positive definite polynomial (or 'shape function') and the resulting algorithm was denoted as the V-s iteration algorithm [10,35,36]. The elements that affect the performance of V-s iterations including the choice of the initial LF V_0, the degree of the new LF V, and the shape function $p(x)$, have been considered to increase the size of the estimation. A higher-degree LF is searched rather than the usual quadratic ones to achieve a richer level set [31]. However, the computational burden dramatically increases with an increase in the system's dimension and/or the degree of the polynomial. Several methods have been proposed to address this problem. For example, a composite LF consisting of lower-degree LFs, such as pointwise maximum and pointwise minimum LFs have been investigated [5,8,37]. In addition, a family of parameter-dependent LFs was used for improved estimation as an alternative to using only one Lyapunov estimate [38]. A systematic method (denoted as R-composition [39]) was utilized for richer-shaped estimation [40] by composing LFs using R-functions. Optimization of the initial LF V_0 is also important. The method in [13] proposed optimizing V_0 to avoid numerical infeasibility in the initial step when the level set is lower than the solver tolerance. Topcu et al. [5] proposed using information from simulations to generate better LF candidates to improve the performance of an estimation algorithm. Regarding the shape function, a quadratic form was customized for better alignment with the shape of the exact ROA [2] or to reflect the relative importance of states in practical problems [41–43]. However, an appropriate shape function is difficult to define without prior knowledge of the system and a systematic approach has not been proposed [5,13,21]. A quadratic form of

$p(x) = x^T M x$ is widely used, with a general assumption of EP at the origin. The shape matrix $M \in \mathbb{R}^{n \times n}$ (n is the system dimension) is a positive definite matrix that is problem-dependent and commonly used to scale the state space domain. In most applications, the shape function is fixed throughout the iterations [44]. In [2], an adaptive shape function was proposed to update the quadratic parts of the newly found LF in each iteration. A better estimation is obtained for some examples, especially for those with a symmetric or simple ROA. However, the same problem of convergence is encountered after certain iterations in the conventional V-s algorithm.

In other ROA estimation methods such as [13], an additional constraint is introduced to improve the estimation, but it can be highly restrictive when a higher-degree LF is searched. Tan [31] proposed the use of a series of consecutive rotations of an elliptical-shaped function for enlargement of ROA estimation. A similar approach was proposed based on the variation of the matrix P [9]. In [45], sufficient conditions were provided to guarantee that the sublevel sets of the polynomial LFs yielded an inner approximation of the exact ROA up to any desired accuracy. However, these conditions are only applicable to the bounded ROA. The application of SOS programming to the Van de Pol system yields a result similar to that obtained in [2]. A simple algorithm for handling the bilinear problem for ROA estimation was described in [22]. However, it is similar to the two-way iterative search in the widely used V-s iteration algorithm.

In addition to the aforementioned modifications, the V-s iteration algorithm uses an increased number of iterations to enlarge the ROA estimation. However, this is not a universal remedy. For example, in cases of early converged optimization and/or numerical infeasibility, particularly for a system with an unbounded or irregular ROA, this approach does not work.

Journal Pre-proof

Another feature is that the geometric center of the shape function, either fixed or adaptive, is located at the origin. This limits the estimation, especially for non-symmetric or unbounded ROA.

The main contribution of this work is a modification of the V-s algorithm that facilitates the use of shifted shape functions (SSF) with shifting centers. This approach yields a significantly enlarged ROA estimation, especially for nonsymmetric or unbounded ROA, as compared to existing methods. To the best of our knowledge, the form of the shifted shape function proposed in this paper has never been explored, although it can be absorbed in the general construction of SOS polynomials. The SSFs are constructed by shifting the centers that are chosen iteratively from the proven obtained level set of the true ROA. Thus, it is advantageous to utilize previously obtained results. The proven level set can be conveniently obtained using the V-s iteration algorithm. In addition, the V-s algorithm allows the SSF to satisfy the constraints. A set of shifting centers with corresponding SSFs is generated to produce several proven subsets of ROA. The union of these sets based on R-composition yields a compact and richer-shaped result without an increase in the LF degree, thus alleviating the computational burden. The proposed approach is denoted as RcomSSF in the remainder of this report. It pushes forward the extension when additional iterations do not provide further ROA expansion or when numerical infeasibility is encountered. In addition, it can exploit the improvements of the V-s iteration algorithm, for instance, modifications of the shape matrix and the initial LF.

The remainder of this paper is organized into different sections. First, the ROA estimation problem is described using fundamental theories in Section 2. Then, using SOS techniques, the computational V-s iteration algorithm is presented. Section 3 describes the shape function as a

premise of the proposed method. The proposed RcomSSF is then presented in Section 4 including the details of the construction and implementation of the SSF and R-composition. Section 5 presents several numerical benchmark examples from the literature. The effectiveness and advantages of the proposed method are demonstrated by comparing it with existing ROA estimation algorithms. Finally, Section 6 summarizes the main conclusions of the report.

2 Region of attraction estimation

2.1 Problem formulation

Consider an autonomous nonlinear polynomial system

$$\dot{x} = f(x), \quad x(0) = x_0, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector and $f(x)$ is a $n \times 1$ polynomial vector field. Without loss of generality, we assume that the origin is an asymptotically stable EP such that $f(0) = 0$. The ROA, a set of initial conditions whose trajectories do not go beyond this set and always converge back to the origin, can be defined as

$$\Omega := \left\{ x_0 \in \mathbb{R}^n : \text{If } x(0) = x_0 \text{ then } \lim_{t \to \infty} x(t) = 0 \right\}. \tag{2}$$

An exact ROA can be obtained only for relatively simple nonlinear systems and is almost impossible for real-world systems. As such, numerous studies have investigated the development of methods for ROA estimation [2,4–20,46]. The estimation of the ROA in this study is based on Lemma 1 using the direct Lyapunov theorem that specifies a sublevel set of a LF as an inner approximation of the true ROA of an asymptotically stable EP [37].

Lemma 1 If there exist a continuously differentiable scalar function $V(x):\mathbb{R}^n \to \mathbb{R}$ and a positive scalar $\gamma \in \mathbb{R}^+$ such that

$$V(x) > 0 \quad \forall x \neq 0 \quad \text{and} \quad V(0) = 0, \tag{3}$$

$$\Omega_\gamma := \{x : V(x) \leq \gamma\} \quad \text{is bounded,} \tag{4}$$

$$\Omega_\gamma \subseteq \{x : (\partial V / \partial x) f < 0\} \cup \{0\}, \tag{5}$$

then the origin is asymptotically stable and Ω_γ is a subset of the ROA. The level γ can be optimized to obtain the largest possible estimation. $V(x)$ that satisfies Lemma 1 is called a strict LF. When γ in Eq. (4) is unbounded, the system is globally asymptotically stable [7].

To enlarge the estimation Ω_γ, an interior expanding algorithm [2,34,44] is introduced, wherein a scalar polynomial function $p(x) \in \boldsymbol{R}[x]$ ($\boldsymbol{R}[x]$ is the set of polynomials with real coefficients in $x \in \mathbb{R}^n$) and another positive scalar $\beta \in \mathbb{R}^+$ are defined. It is ensured that

$$\varepsilon_\beta := \{x \in \mathbb{R}^n : p(x) \leq \beta\} \quad \text{is bounded,} \tag{6}$$

$$\varepsilon_\beta \subseteq \Omega_\gamma, \tag{7}$$

where $p(x)$ which denotes the shape function is a positive definite and convex polynomial. The positive scalar β is maximized by imposing constraints (3)–(5).

The polynomial constraints can be relaxed into SOS constraints using SOS techniques based on the connection between the nonnegativity and the sum of squares [7,20] and then solved using SOS programming. For the set containment constraints in Eqs. (5) and (7), a well-known generalized S-procedure [5,7] is employed.

Lemma 2 (generalized S-procedure [5]). Given polynomials $g_0(x), g_1(x) \ldots g_m(x) \in \boldsymbol{R}[x]$, $x \in \mathbb{R}^n$ and $s_1(x) \ldots s_m(x) \in \Sigma_n$, if

$$g_0(x)-\sum_{i=1}^{m}s_i(x)g_i(x)\geq 0\,, \tag{8}$$

then

$$\left\{x\in\mathbb{R}^n : g_1(x),g_2(x)...g_m(x)\geq 0\right\}\subseteq\left\{x\in\mathbb{R}^n : g_0(x)\geq 0\right\}. \tag{9}$$

Therefore, the set containment constraint in Eq. (5) can be formulated as follows

$$-\left[(\partial V/\partial x)f+l_2\right]-(\gamma-V)s_2\geq 0\,, \tag{10}$$

where Σ_n denotes the set of SOS polynomials in $x\in\mathbb{R}^n$. Constraints (7) can be reformulated in the same manner. Therefore the ROA estimation problem (3)–(7) can be formulated as an optimization problem

$$\begin{aligned} &\underset{s_1,s_2\in\Sigma_n}{\boldsymbol{max}}\ \beta \\ \text{Subject to}\quad & V-l_1\in\Sigma_n \\ & -\left[(\partial V/\partial x)f+l_2\right]-(\gamma-V)s_2\in\Sigma_n \\ & (\gamma-V)-(\beta-p)s_1\in\Sigma_n \end{aligned}, \tag{11}$$

where $l_i(x)(i=1,2)$ is a small positive polynomial (typically $\varepsilon x^T x$ with some small $\varepsilon\in\mathbb{R}^+$) to guarantee that V is strictly positive and the derivative of V is strictly negative. $s_i(x)(i=1,2)$ is an auxiliary SOS multiplier with an appropriate degree. The resulting optimization problem (11) is a bilinear problem with β, γ, and the decision variables in V coupled with that in s_i.

2.2 V-s **iteration algorithm**

A straightforward way of solving the bilinear optimization problem (11) is to use a bilinear solver, which is not as well established as linear solvers [31]. A bypass solution is to relax the problem into linear subproblems, followed by the use of a two-way iterative search algorithm, which leads to the so-called V-s iteration algorithm [31,44]. The problem (11) can then be

Journal Pre-proof

formulated as a three-linear problem with three steps: γ-step, β-step, and V-step. When the stopping criteria of the algorithm are met, a new normalized LF V^*, is obtained with a proven level set

$$\Omega^* := \left\{ x \in \mathbb{R}^n : V^* < 1 \right\}, \tag{12}$$

where V^* is the optimized LF and Ω^* is the largest possible level set taken as an estimate of the true ROA. Given an initial feasible LF, V_0, a shape function p_0, and the expected number of iterations N, the V-s iteration algorithm, denoted as Algorithm 1, can be executed as follows

Algorithm 1: V-s **iteration**

Input: an appropriate LF $V_0(x)$; a shape function $p_0(x)$; the number of iterations N;

Output: V^*.

1: $V = V_0, \quad p = p_0;$

2: **for** $i = 1:N$ **do**

3: γ**-step: hold** V **fixed and solve for** s_2 **and** γ^***:**

$$\gamma^* := \max_{s_2 \in \mathrm{SOS}} \gamma \quad \textbf{s.t.} \quad -[(\partial V / \partial x) f + l_2] - (\gamma - V) s_2 \in \Sigma_n; \tag{13}$$

4: β**-step: hold** V **and** γ^* **fixed and solve for** s_1 **and** β^***:**

$$\beta^* := \max_{s_1 \in \mathrm{SOS}} \beta \quad \textbf{s.t.} \quad (\gamma^* - V) - (\beta - p) s_1 \in \Sigma_n; \tag{14}$$

5: $V_{old} = V, \quad \beta^*_{old} := \beta^*;$

6: V**-step: hold** $s_1, s_2, \beta^*, \gamma^*$ **fixed and solve for the new** V **that satisfies:**

$$\begin{aligned} -[(\partial V / \partial x) f + l_2] - (\gamma^* - V) s_2 &\in \Sigma_n \\ (\gamma^* - V) - (\beta^* - p) s_1 &\in \Sigma_n \\ V - l_1 &\in \Sigma_n \\ V(0) &= 0 \end{aligned}; \tag{15}$$

7: **if** (15) **is feasible then**

$V = V / \gamma^*$,

else

$V = V_{old} / \gamma^*$,

return

end if

8: **if** $\left|\left(\beta^*_{old} - \beta^*\right) / \beta^*\right| < \varepsilon_{TOL}$ **then**

return

end if

9: **end for**

10: $V^* = V$, $\Omega^* := \left\{x \in \mathbb{R}^n : V^* < 1\right\}$.

Remark 1: To satisfy Eq.(14), the degree of the SOS multiplier s_1 is chosen such that $\deg p + \deg s_1 \geq \deg V$. In Eq. (13), $f(0) = 0$ indicates that there are no constant terms in $(\partial V / \partial x) f$ so the multiplier s_2 associated with the term $(\gamma - V)$ in Eq. (13) should not include a constant term. The degree of s_2 is then chosen to be larger than the maximum degree of s_2, l_2, and $(\partial V / \partial x) f$. Additional details on the practical aspects of the computation process can be found in the literature [2,31].

Remark 2: The choice of the initial LF V_0 is flexible if it guarantees that the initial search is feasible. An optimized V_0 is favorable for better estimation; for instance, using the optimization procedure in [13] or the simulation-guided procedure in [3] to obtain a better V_0. However, considering the difficulty of constructing a LF, a more systematic method is widely used in the literature as follows

$$V_0 = x^T P x, \tag{16}$$

where P is computed using the Lyapunov equation

$$A^T P + P A = -Q, \tag{17}$$

Journal Pre-proof

where $A = (\partial f / \partial x)\big|_{x=0}$ is considered Hurwitz. P is a positive definite matrix. $Q > 0$, and $Q = I$ (I is the unit matrix) is typically selected to allow for the largest ball estimation [2].

Remark 3: In addition to expanding the estimation from the interior, the shape function $p(x)$ can be used to measure the size of the ROA. The convergence of β and the other elements can be considered as the stopping criteria of the algorithm, such as

A. two consecutive β are less than a programmed tolerance ε_{TOL};

B. numerical infeasibility alert for the optimization problem in Eqs. (13)–(15);

C. the specified number of iterations N is reached.

Remark 4: The V-s iteration algorithm converts the bilinear problem into an iteration procedure that includes two optimization problems (γ-step and β-step) and a feasibility problem (V-step) at each iteration, which can be solved using linear SDP solvers, such as SeDuMi. The bisection procedure must be applied in the γ-step and β-step because the optimization variable γ (β) is coupled with the decision variables in s_2 (s_1).

Remark 5: The computational complexity increases dramatically with the scale of the problem, namely, with the degree and dimension of the system $f(x)$ and the degree of the LF that is searched. This is an inherent limitation of SOS optimization. Thus, the approach is limited to systems with fewer than eight to ten states with a cubic degree. Polynomial models of higher degrees can be addressed with fewer states [47]. Usually, the γ-step is responsible for much of the computational burden owing to the higher overall degree. Suitable software and additional information about the V–s iteration can be found in [48]. The free-distributed toolbox SOSOPTS [26] is used in this study.

Journal Pre-proof

3 Preliminaries on shape function

As mentioned in Section 2.1, a user-defined shape function $p(x)$ is introduced to enlarge the inner approximation of the ROA, and an appropriate choice of $p(x)$ can enhance the estimation [2]. According to this definition, $p(x)$ can be any positive definite and convex polynomial, but a widely used choice is a quadratic function with its geometric center at the origin

$$p(x) = x^T M x, \tag{18}$$

where the positive definite matrix M is used for practical implementation. This form is chosen for three reasons. First, the introduction of the β-step incurs additional computational cost. Therefore, the interior expansion step is implemented using a computationally cost-efficient quadratic form to avoid obscuring the estimation improvement owing to the incurred computation cost. Second, the system is assumed to have an asymptotically stable EP at the origin. Thus, the origin is the only identified point inside the true ROA without additional knowledge, which justifies the selection of the geometric center at the origin. Third, the shape matrix M is vital for an accurate estimation because it is associated with physical meaning, such as the shape information of the ROA, dimensional scaling information, and the importance of certain directions in the state space [44]. Its choice is problem-dependent, but generally, M is used as an identity matrix when prior knowledge is unavailable. The quadratic V_0, which is computed using Eq. (16), can also be considered as a shape function.

In the traditional V-s iteration (**Algorithm 1**), the shape function is fixed. A modification in [2] proposed the updating of the shape function iteratively using the quadratic component of a

newly found LF. This modified algorithm (**Algorithm 2**) can be illustrated by an additional step after the V -step in **Algorithm 1**.

Algorithm 2: Modified V-s **iteration using an adaptive shape function**	
1-7:	Same as Steps 1–7 in **Algorithm 1**;
7a:	Update the shape function $p(x)$ using the quadratic component of V ;
8-10:	Same as Steps 8–10 in **Algorithm 1**.

Given that V must be positive definite, and $V(0) = 0$, it does not consist of constant and linear terms. Therefore, the adaptive shape function will only have quadratic terms, which implies that the geometric center of the shape function is still at its origin. As demonstrated in [2], the adaptive shape function aligns better with a simple-shaped ROA, for instance, the ROA of the Van de Pol system. However, this is challenging for a complex-shaped or unbounded ROA. In addition, it aligns with only one direction (or converges in one direction) and possibly causes estimation degradation in other directions. Moreover, it suffers from the same convergence problem as **Algorithm 1**.

For brevity, **Algorithms 1** and **2** will be designated as A1 and A2, respectively, throughout the report.

4 ROA estimation via RcomSSF

Although the algorithm with the previously mentioned modifications could improve ROA estimations in some cases, it still provides conservative estimations and requires significant computational resources. For the V-s iteration algorithm, with the geometric center of the shape function located at the origin, the expansion is bounded in a domain around the origin. Thus, its application is limited for nonsymmetric and unbounded ROA. In this study, the proposed

Journal Pre-proof

RcomSSF improves ROA estimation using SSFs and exploits the algorithm without being hindered by its limitations.

4.1 **Shifted shape function**

This section examines the concept of the SSF. Rather than being fixed at the origin, as in the conventional shape function in Eq. (18), the center of the shape function is shifted away from the origin to another point inside the verified level set. Given that the center is a valid point inside the true ROA, the SSF defined in this way complies with the definition of the shape function. It plays the role of interior expansion and guides the estimation growth towards a new region. The SSF is effective, especially for non-symmetric or unbounded ROA, because of the offset of the shifting center. The shifts can be performed for any shape function; however, for simplicity in demonstration and implementation, an ellipsoidal shape function with a shifting center x^* is considered in this study

$$p(x) = (x - x^*)^T M (x - x^*) . \tag{19}$$

This can be considered as an extension of Eq. (18) with additional constant and linear terms introduced by x^*. Thus, it can be incorporated into the general construction of SOS polynomials. Although it is simple, this definition of shape function has never been explored, to the best of the author's knowledge. This form, compared with the general SOS construction using monomials, is an efficient and systematic way to construct shape functions using x^* and M. This method of construction allows explicit physical information or expert knowledge to be embedded into the process to improve the ROA expansion capabilities. In addition, because the proposed shape function construction fits the SOS framework, elaborate numerical optimization procedures facilitate this approach.

Journal Pre-proof

The shape functions with different shifting centers and shape matrices are illustrated using two-dimensional examples in Fig. 1. A comparison between $p_2(x)$ and $p_1(x)$ shows the effect of different shape matrices, whereas a comparison between $p_3(x)$ and $p_2(x)$ shows the shifting effect of x^*.

4.2 **Rounds of shifts**

The shifting procedure implemented in the shift rounds is described in this section. A shape function with a new center x^* will generally produce a new level set Ω^*. In the next round, a new center is selected inside the level set obtained in the previous round. As such, a series of shifts yields a series of different proven level sets via iterative shifting. This process can be continued until convergence to the exact ROA occurs. Subsequently, these acquired level sets are combined into one set via R-composition [39,40] as the final verified inner approximation of the exact ROA.

The algorithm is discussed in detail in the following. Initially, the first attempt at a proven level set of the ROA is performed, for example, using Eq. (16). Subsequently, V-s iteration is used to obtain a new improved estimation $\Omega_0^* := \left\{x \in \mathbb{R}^n : V_0^* < 1\right\}$. A1, A2, or other algorithms can then be used to find Ω_0^*, which is used by RcomSSF as an initial approximation. Subsequently, using the shifting procedure, other proven level sets stem from Ω_0^*.

As shown in Fig. 2, a set of shifting centers x_i^* ($i = 1,2,...$, i indexes the shifts in the first round) is chosen from Ω_0^* for the first round of shifts. Then, the V-s algorithm is applied to obtain the corresponding optimized LF V_i^* and the relevant proven level sets $\Omega_i^* := \left\{x \in \mathbb{R}^n : V_i^* < 1\right\}$. Next, for the second round of shifts, x_{ij}^* ($j = 1,2,...$ index shifts in

the second round) are chosen inside the respective proven level set Ω_i^* and produce the second-generation optimized LFs V_{ij}^*, and the corresponding proven level sets $\Omega_{ij}^* := \{x \in \mathbb{R}^n : V_{ij}^* < 1\}$. Similarly, the third round of shifts is performed to obtain the third generation of proven level sets $\Omega_{ijk}^* := \{x \in \mathbb{R}^n : V_{ijk}^* < 1\}$ ($k = 1,2,...$ indexes the shifts in the third round). The tree structure for this process is illustrated in Fig. 2. To differentiate different shifts, the indexes i, j, k denote the independent shifts in the first, second, and third rounds, respectively. The subscripts, except for the last one, refer to the parent LF and the parent level set; for instance, $\Omega_{123}^* := \{x \in \mathbb{R}^n : V_{123}^* < 1\}$ is obtained for the center x_{123}^* and is chosen from the parent level set $\Omega_{12}^* := \{x \in \mathbb{R}^n : V_{12}^* < 1\}$. For succinctness, the approach is demonstrated for only three rounds of shifts; however, without loss of generality, it can be extended to a higher number of rounds.

It should be mentioned that the V-s iteration algorithm for the SSF must have a fixed shifting center to maintain expansion around the specific center. Otherwise, the shifting effect will be neutralized. For example, if A2 is used, the adaptive shape function composed of the quadratic terms of the new LF will always maintain the center at the origin. The examples in Section 5 are used to demonstrate the algorithm.

4.3 **Selection of shifting centers**

Here some considerations for selecting the shifting center x^* are considered.

First, x^* must be contained within a proven level set Ω to guarantee the set containment constraint in Eq. (7) is satisfied, thus ensuring that the following iterations are feasible.

Second, x^* should be close to the boundary of Ω. For clarity, a two-dimensional example is used for the demonstration in Fig. 3. Several locations (denoted by C_i and $i = 1,2,...$) of

Journal Pre-proof

x^* are shown in Ω. C_1 is expected to yield a larger ROA estimation than C_2, because C_1 is farther from the origin and closer to the boundary. One method involves manually selecting the location of x^* in a proven level set after the previous estimation round. However, for general cases, a more generalized procedure may be required. Here, we propose a method for constructing a general selection algorithm. The algorithm is illustrated in Fig. 3. First, a phase-direction of the ROA expansion should be specified. This direction can be determined either by iteratively sweeping the entire domain or by prioritizing particular directions based on expert knowledge. Subsequently, a line can be assigned to each given phase angle α. The intersection of each line with the available level set boundary curve defines the point P_α. The coordinates of P_α can be obtained by simultaneously solving the equations of the line and level curves. Subsequently, the distance from the origin to the intersection point P_α can be determined. As such, the shifting center can be chosen as $x^* = \sigma \cdot x_{P_\alpha}$, where x_{P_α} is the coordinate of , and $\sigma \in (0,1)$ is the design parameter. The simulations show that a higher number of iterations is required for a smaller value of σ to push the estimation beyond the parent level set. Otherwise, for a larger σ, the center is too close to the boundary, and this proximity results in a small β in Eq. (6) thereby reducing the capability of the interior expansion algorithm. For the cases considered in this study, the optimal value of σ is 0.8.

Third, a center x^* near the convex boundary usually yields a better estimation. In contrast, a center near the concave boundary sees little estimation expansion because it has already approached the concave boundary of the exact ROA. For example, the case considered in Fig. 3, $C_i (i = 1,4,5,6)$ is superior to C_3.

Fourth, there is no limit on the number of selected centers, and in general, the higher the number, the more accurate the estimation. However, it should be noted that increasing the number of selected centers also increases the computational cost. Therefore, the selection of an appropriate number of centers is a trade-off between accuracy and computational cost.

The aforementioned considerations will be discussed further in Sec. 5 based on simulation examples.

4.4 **R-composition**

R-composition is a systematic way of composing LFs using R-functions. It facilitates the acquisition of richer and more flexible LFs and has been used for ROA estimation [39,40]. In this investigation, it is employed in the final step of the RcomSSF to provide a more compact result by combining the independent level sets obtained from the shifting procedure into a single set. The compact form of the result is also beneficial for further application. Thus, the proposed method is more complete.

R-functions can represent the natural extension of Boolean operators (e.g., AND, OR, NOT) for real-valued functions and are, therefore, essential tools for representing the intersection, union, and complement in a geometric setting. The full description of R-functions is beyond the scope of this work and can be found in [39] and the references presented therein.

Constructed using the LF $V^*(x)$, the function

$$R(x) = 1 - V^*(x) \tag{20}$$

is an R-function. When $R(x) > 0$, the set $\hat{R} = \left\{ x \in \mathbb{R}^n : R(x) > 0 \right\}$ is the proven level set in Eq. (12) for ROA estimation. The Boolean functions “NOT,” “AND,” and “OR” correspond to the R-functions, “R-negation,” “R-intersection” and “R-union” in Table I.

Journal Pre-proof

The R-composition parameter τ from Table I which is chosen within $(0, 2]$, facilitates some implementation freedom in this work, and $\tau = 2$ is chosen. Geometrically, if $R_1(x)$ and $R_2(x)$ are positive inside a geometrical region and negative outside, then the R-intersection and R-union represent the intersection and union between the two sets $\hat{R}_1 := \{x \in \mathbb{R}^n : R_1(x) > 0\}$ and $\hat{R}_2 := \{x \in \mathbb{R}^n : R_2(x) > 0\}$, such that

$$\hat{R}_1 \cap \hat{R}_2 = \{x \in \mathbb{R}^n : R_\cap(R_1, R_2) > 0\}, \quad \hat{R}_1 \cup \hat{R}_2 = \{x \in \mathbb{R}^n : R_\cup(R_1, R_2) > 0\}. \tag{21}$$

For instance, if $R_1(x) = 1 - x^T M_1 x$ and $R_2(x) = 1 - x^T M_2 x$, where $M_1 = \text{diag}(1,9)$ and $M_2 = \text{diag}(9,1)$, then the intersection and the union between sets $\hat{R}_1$ and $\hat{R}_2$ are shown in Fig. 4.

Therefore, the R-union is adopted in RcomSSF to produce the union of the estimation. For example, the union of the ROAs corresponding to the LFs $V_i^*, V_{ij}^*, V_{ijk}^*, ...(i, j, k = 1,2,3...)$ that are obtained via rounds of shifts can be computed iteratively as follows

$$\Omega_e = \{x \in \mathbb{R}^n : R_e > 0\}, \quad R_e = R_\cup(R_\cup(R_\cup(1 - V_0^*, 1 - V_i^*), 1 - V_{ij}^*), 1 - V_{ijk}^*, ...) \tag{22}$$

and we have

$$\Omega_e = \Omega_0^* \cup \Omega_i^* \cup \Omega_{ij}^* \cup \Omega_{ijk}^* \cdots \tag{23}$$

It is considered in [39,40] that $-R_e(x) + R_e(0)$ is a LF when $\tau = 2$ or, more precisely, a Lyapunov-like function, because the classical LF condition that requires continuous differentiability is relaxed. The sufficient conditions for $-R_e(x) + R_e(0)$ are that a LF can be found in [39,40]. A detailed discussion is omitted because R-composition in RcomSSF assumes the task of better communication of the result in a compact form, and the union of the LFs is not used to initialize the next optimization process, although this is possible.

4.5 **RcomSSF**

Journal Pre-proof

Based on the aforementioned details, the final RcomSSF algorithm can be summarized as follows

Algorithm 3: RcomSSF

Input: an appropriate LF, $V_0(x)$; shape function $p_0(x)$; the number of iterations N

Output: Ω^*

1: Run V-s iteration to obtain the optimized LF, V_0^* and the relevant parent level set $\Omega_0^* := \{x \in \mathbb{R}^n : V_0^* < 1\}$;

2: The first round of shifts:

Initialization:

Give phase angle and coefficient σ to calculate ρ_α and locate shifting centers x_i^* ($i = 1,2,...$) in Ω_0^*; specify the shape matrix M; $V_0 = V_0^*$, $p_0 = (x - x_i^*)^T M (x - x_i^*)$ and iteration number N;

Run V-s **iteration:** obtain the optimized LF V_i^* and level set Ω_i^***;**

Further shift check:

Calculate ρ_α again as $\rho_{\alpha New}$

if $\left|(\rho_{\alpha New} - \rho_\alpha) / \rho_\alpha\right| > \varepsilon_{TOL\rho}$ **then**

continue the next round;

else

go to Step 6;

end if

3: The second round of shifts:

Initialization:

Give phase angle and coefficient σ to calculate ρ_α and locate shifting centers x_{ij}^* ($j = 1,2,...$) in $\Omega_i^* := \{x \in \mathbb{R}^n : V_i^* < 1\}$; specify the shape matrix M; $V_0 = V_i^*$, $p_0 = (x - x_{ij}^*)^T M (x - x_{ij}^*)$ and iteration number N;

Run V-s **iteration:** obtain the optimized LF, V_{ij}^* and the level set Ω_{ij}^*

Further shift check.

Journal Pre-proof

4: The third round of shifts:

Initialization:

Give phase angle and coefficient σ to calculate ρ_α and locate shifting centers $x_{ijk}^{*}\,(k=1,2,...)$ in $\Omega_{ij}^{*} := \left\{x \in \mathbb{R}^n : V_{ij}^{*} < 1\right\}$; specify the shape matrix M; $V_0 = V_{ij}^{*}$, $p_0 = (x - x_{ijk}^{*})^T M (x - x_{ijk}^{*})$ and iteration number N;

Run V-s **iteration (A1):** obtain the optimized LF, V_{ijk}^{*} and level set Ω_{ijk}^{*};

Further shift check.

5: The fourth round of shifts …

6: R-composition of Ω_0^{*}, Ω_i^{*}, Ω_{ij}^{*}, and Ω_{ijk}^{*} …into one single level set Ω_e using Eq. (22).

The proximity of the center to the boundary $\left|(\rho_{\alpha New} - \rho_\alpha)/\rho_\alpha\right|$ is chosen as a metric for further shift checks. The tolerance $\varepsilon_{TOL\rho}$ can be customized to 10%, which means that if the expansion in the phase direction is less than 10%, then the shifting stops in this direction; otherwise, a new center is decided and the next round of shifts occurs. Shifts in the same round can be performed in parallel to reduce the entire verification period.

In RcomSSF, an increase in the number of shift rounds with an increased number of selected shift centers leads to a linear increase in the number of iterations. This is advantageous in terms of the computation cost compared to other modifications, for example, improvement by using an increased degree of LF. Moreover, the shifting procedure yields more significant growth in the estimation. These considerations account for the computational efficiency of the proposed method. In addition, when applied to practical problems, RcomSSF is feasible provided that the V-s iteration algorithm is feasible, and RcomSSF yields a better estimation without the implementation of more sophisticated approaches.

Generally, prior expert knowledge of the true ROA can facilitate other algorithms by specifying the N matrix of the shape function[3,13,40,44]. However, RcomSSF is effective even without expert knowledge of the shape functions or locations of the shifting center.

In the next section, RcomSSF is benchmarked against A1, A2, and other methods from the literature using several examples.

5 Simulations

5.1 Systems with bounded ROAs

Let us consider systems with bounded ROAs.

Example 1. Here we consider a Van de Pol system taken from [2]

$$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 + 5x_2(x_1^2 - 1) \end{cases}. \tag{24}$$

It has a stable EP at the origin and an unstable limit cycle. The problem of ROA estimation of the Van de Pol system has been extensively studied [2,3,6,38,49], and it is considered a benchmark example for RcomSSF validation. Its exact ROA is the region enclosed by the limit cycle, which can be plotted using the reverse trajectory method, as shown in Fig. 5 (a). To begin the estimation, the initial LF $V_0(x)$ is computed using Eq. (16) for $Q = I$, and the matrices A and P are defined as follows

$$A = \begin{bmatrix} 0 & -1 \\ 1 & -5 \end{bmatrix}, \quad P = \begin{bmatrix} 2.7 & -0.5 \\ -0.5 & 0.2 \end{bmatrix}.$$

The computation is then initialized based on the following expressions

$$V_0(x) = 2.7x_1^2 - x_1x_2 + 0.2x_2^2, \quad p_0(x) = V_0(x). \tag{25}$$

Subsequently, a new six-degree LF is investigated in the V -step. The RcomSSF prediction of the ROA is shown in Fig. 5. The results obtained for algorithms A1 and A2 in [2] under the same initial conditions are also shown for comparison. Fig. 5 (a) shows that when a six-degree LF is searched using the iteration number $N = 30$ and 60, A2 with an adaptive shape function yields a larger estimation than A1 with a fixed shape function. However, it still fails to predict the exact ROA. Moreover, there is no improvement in the estimation for both A1 and A2 after 20 iterations because the algorithm converges. This is examined in further detail in Fig. 5 (b) for A2. The variation in the adaptive shape function is plotted for the 1st, 30th, and 60th iterations. The results indicate that the ellipse rotates anticlockwise to align better with the shape of the true ROA. Stretching along the major axis and shrinking along the minor axis can also be observed, which subsequently leads to a small increase and decrease in the estimation in the corresponding directions. However, the significant rotation and stretching almost stop after 30 iterations, and the estimation process converges. In summary, A1 yields worse estimations compared to A2 for the same number of iterations. Both A1 and A2 converge before reaching the true ROA; therefore, a further increase in the number of iterations does not contribute to the expansion of the estimate. Shrinking of the shape function in a certain direction leads to a worse estimation.

The proposed RcomSSF addresses these issues. The V-s iteration is performed first to yield a valid initial level set. The final V^* for A1 at $N = 30$ is selected as the initial LF V_0^*. A shape matrix $M = I$ is selected for the general circle $p(x)$. To initialize the first round of shifts, two centers $x_1^* = [1, 1]$ and $x_2^* = [-1, -1]$ are selected directionally inside the level set

$\Omega_0^* := \{x \in \mathbb{R}^n : V_0^* < 1\}$, given the gap between the exact ROA. Subsequently, the shape function is constructed using Eq. (19) with $x_1^* = [1, 1]$ as follows

$$p_1(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2.$$

Using this shape function, the proven level set $\Omega_1^* := \{x \in \mathbb{R}^n : V_1^* < 1\}$ (green dotted line in Fig. 5 (c)) that very precisely covers the ROA for $x_2 > 0$ is obtained. Another shifting center $x_2^* = [-1, -1]$ gives the second shape function

$$p_2(x) = x_1^2 + x_2^2 + 2x_1 + 2x_2 + 2.$$

Likewise, using the second shape function, the proven level set $\Omega_2^* := \{x \in \mathbb{R}^n : V_2^* < 1\}$ (green dashed line in Fig. 5 (c)) that precisely covers the ROA for $x_2 < 0$ is obtained. Together, the union of the three level sets Ω_0^*, Ω_1^*, and Ω_2^* is computed using R-composition based on Eq. (22)

$$\Omega_e := \{x \in \mathbb{R}^n \mid R_e > 0\}, \quad R_e = R_U(R_U(1 - V_0^*, 1 - V_1^*), 1 - V_2^*) \tag{26}$$

and

$$\Omega_e = \Omega_0^* \bigcup \Omega_1^* \bigcup \Omega_2^*. \tag{27}$$

As indicated in Table I, R_e is a complex function. However, in this case, a polynomial approximation can be obtained

$$\begin{aligned} R_e(x) = & -0.097268x_1^6 - 0.047707x_1^5x_2 - 0.007790x_1^4x_2^2 + 0.003634x_1^3x_2^3 \\ & -0.001048x_1^2x_2^4 + 0.000208x_1x_2^5 - 3.601840\mathrm{e}^{-5}x_2^6 + 0.000273x_1^5 \\ & -0.000271x_1^4x_2 + 0.000396x_1^3x_2^2 - 9.588542\mathrm{e}^{-5}x_1^2x_2^3 + 1.126447\mathrm{e}^{-5}x_1x_2^4 \\ & -2.250610\mathrm{e}^{-7}x_2^5 + 0.542801x_1^4 + 0.073756x_1^3x_2 - 0.082728x_1^2x_2^2 \\ & +0.021571x_1x_2^3 - 0.003026x_2^4 - 0.000389x_1^3 - 0.000149x_1^2x_2 \\ & -0.000249x_1x_2^2 + 8.912165\mathrm{e}^{-5}x_2^3 - 1.182416x_1^2 + 0.417432x_1x_2 \\ & -0.076201x_2^2 + 1 \end{aligned}.$$

A numerical method with conjectured convergence was proposed and applied to the Van der Pol system in [45]. However, the ROA estimate obtained for a six-degree LF was less than the true ROA and was similar to the A2 prediction shown in Fig. 5 (a). In addition, the method [45] is limited to bounded ROA, however, the proposed RcomSSF can also be applied to the unbounded ROA, as illustrated in the following examples.

5.2 **Systems with unbounded ROAs**

Many real-world systems have unbounded ROAs. These are considered below to validate RcomSSF.

Example 2. Consider the following system [2,50]

$$\begin{cases} \dot{x}_1 = -4x_1^3 + 6x_1^2 - 2x_1, \\ \dot{x}_2 = -2x_2. \end{cases} \tag{28}$$

Analysis of the linearized system, combined with the vector field plot in Fig. 6 (a), shows that the system has three EPs, namely, two stable node sinks (0,0) and (1,0) and the saddle point (0.5, 0). A line $x_1 = 0.5$ divides the plane. The trajectories originating from the left-hand/ right-hand side of this line sink down to the node (0, 0) /(1,0). Therefore, two ROAs correspond to two stable node sinks, and each of them can be treated equivalently. For the sake of brevity, the performance of the RcomSSF will be demonstrated for only one EP, namely, (0,0), with an unbounded ROA in the $x_1 < 0.5$ plane.

For ROA estimation, the initial LF is computed using Eq. (16) for $Q = I$, and the matrices A and P are obtained

$$A = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}, \quad P = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}.$$

Then, the computation is initialized as follows

Journal Pre-proof

$$V_0(x) = 0.25x_1^{\,2} + 0.25x_2^{\,2}. \tag{29}$$

We first examine the results for algorithms A1 and A2 [2,50]. In these two cases, the initial shape function is set as

$$p_0(x) = 0.8\,V_0(x) \tag{30}$$

and the quartic LF is searched. As shown in Fig. 6 (a), A1 converges after approximately 30 iterations. The adaptive shape function takes the form

$$p(x) = 2.58x_1^{\,2} + 3.47\mathrm{e}^{-2}x_2^{\,2} - 3.28\mathrm{e}^{-8}x_1x_2 \tag{31}$$

after the 30th iteration, and A2 covers a much larger area as shown in Fig. 6 (a). However, the simulation result in [2] shows that the left boundary of x_1 is still limited by -1.0, even when the number of iterations is increased to 150. This limitation, however, can be circumvented using the proposed RcomSSF, which allows for an extension in the $-x_1$ direction when using a SSF. The level set obtained for A2 $\Omega_0^* \coloneqq \{x \in \mathbb{R}^n : V_0^* < 1\}$ after 30 iterations is used to produce shifting centers, and a center $x_1^* = [-0.8, 0]$ is chosen to expand the estimate in the $-x_1$ direction. For the shape matrix $M = I$, the SSF constructed using Eq. (19) is given by

$$p_1(x) = x_1^{\,2} + x_2^{\,2} + 1.60x_1 + 0.64. \tag{32}$$

Using RcomSSF, a new proven level set $\Omega_1^* \coloneqq \{x \in \mathbb{R}^n : V_1^* < 1\}$ (green dotted line in Fig. 6 (b)) is obtained. The left boundary is then extended from -1.0 to -1.8, but the estimation along the x_2 axis shrinks. Inspired by the adaptive shape function in A2, a shape matrix $M_1 = \mathrm{diag}(1, 1/16)$ is customized to apply a greater weight to the x_2 axis. In this case, the following shape function is obtained

$$p_{1_{M1}}(x) = x_1^{\,2} + 0.0625x_2^{\,2} + 1.60x_1 + 0.64. \tag{33}$$

Journal Pre-proof

As expected, M_1 produces a larger level set after 30 iterations (green dash-dot line) and after 60 iterations (green dashed line, denoted as $\Omega_{1_{M1}}^* := \left\{x \in \mathbb{R}^n : V_{1_{M1}}^* < 1\right\}$). The estimation expands along not only the $-x_1$ axis but also the x_2 axis. The union of the obtained level sets is computed using Eq. (22) as

$$\Omega_e := \{x \in \mathbb{R}^n \mid R_e > 0\}, \quad R_e = R_{\cup}(1 - V_0^*, 1 - V_{1_{M1}}^*) \tag{34}$$

and

$$\Omega_e = \Omega_0^* \bigcup \Omega_{1_{M1}}^*. \tag{35}$$

The resulting level set is represented by the red dotted line in Fig. 6 (b).

This example shows the ability of the proposed RcomSSF method to extend the estimation in a specific direction defined by a shifting center.

Example 3. Consider the following system [2]

$$\begin{cases} \dot{x}_1 = -50x_1 - 16x_2 + 13.8x_1x_2 \\ \dot{x}_2 = 13x_1 - 9x_2 + 5.5x_1x_2 \end{cases}. \tag{36}$$

Analysis of the linearized system yields a stable node (0, 0) and a saddle point (1.45, 18.17). The vector field plot in Fig. 7 (a) shows the boundary of the ROA for EP (0, 0). At the initialization stage of the algorithm, $V_0(x)$ is computed using Eq. (16) for $Q = I$ as follows

$$V_0(x) = 0.011694x_1^2 + 0.013034x_1x_2 + 0.043969x_2^2. \tag{37}$$

The shape function is set to $p_0(x) = V_0(x)$, and a four-degree LF is searched iteratively.

In Fig. 7 (a), the results of A1 (blue lines) and A2 (black lines) are shown. There is no significant improvement after 30 iterations, which indicates that convergence is achieved. This reinforces the idea that an increase in the number of iterations cannot be a universal remedy. Moreover, the estimation obtained via A1 approaches the exact boundary only in a specific

region, and the adaptive shape function in A2 guides the estimation to grow in a skewed direction; it does not fit the boundary. However, an unbounded ROA means that there is an opportunity for estimation growth. The proposed RcomSSF fulfills this extension using the shifting procedure. The final ROA estimate obtained after two rounds of shifting is plotted in Fig. 7 (a). The results show that the proposed algorithm significantly improves ROA prediction.

The effects of the shape function parameter, namely, centers and N matrix, on the estimation results and the algorithm application are discussed in the following. For the first round of shifts, the proven level set obtained after 30 iterations of A1 is chosen as the parent level set $\Omega_0^* := \{x \in \mathbb{R}^n : V_0^* < 1\}$. Two different centers $x_1^* = [0, -4]$ and $x_2^* = [-7.5, 0]$ are chosen to pull the estimation towards a negative x_1 and a negative x_2. For the center $x_1^* = [0, -4]$, the shape matrix $M_1 = \text{diag}(1/4, 1)$ is selected. The corresponding shape functions are as follows

$$p_1(x) = 0.25x_1^2 + x_2^2 + 8x_2 + 16. \tag{38}$$

For the center $x_2^* = [-7.5, 0]$, the identity matrix is selected as a shape matrix, and the shape function is as follows

$$p_1(x) = x_1^2 + x_2^2 + 15x_2 + 56.25. \tag{39}$$

The obtained proven level sets are identified using green lines (green solid line $\Omega_1^* := \{x \in \mathbb{R}^n : V_1^* < 1\}$ and green dashed line $\Omega_2^* := \{x \in \mathbb{R}^n : V_2^* < 1\}$) in Fig. 7 (b). Even the first round of shifting significantly extends the ROA estimation. The second round of shifts is then performed. Three new centers are selected from the first-round level set Ω_1^*. The center $x_{11}^* = [0, -11]$ is selected to move the extension further into the third and fourth quadrants. For the shape matrix, $M_1 = \text{diag}(1/4, 1)$, the SSF is given as

$$p_{11}(x) = 0.25x_1^{\,2} + x_2^{\,2} + 22x_2 + 121 \tag{40}$$

The resulting level set $\Omega_{11}^{\,*} := \left\{x \in \mathbb{R}^n : V_{11}^{\,*} < 1\right\}$ is represented by the magenta solid line in Fig. 7 (b). Then, in $\Omega_2^{\,*}$, the centers $x_{21}^{\,*} = [-18, 2]$ and $x_{22}^{\,*} = [-3, 8]$ are chosen to extract the estimation in the second and third quadrants, respectively. For $M = I$, the SSFs are obtained as follows

$$p_{21}(x) = x_1^{\,2} + x_2^{\,2} + 36x_1 - 4x_2 + 328\,, \quad p_{22}(x) = x_1^{\,2} + x_2^{\,2} + 6x_1 - 16x_2 + 73\,. \tag{41}$$

The resulting level sets are represented in Fig. 7 (b) by the magenta dashed line ($\Omega_{21}^{\,*} := \left\{x \in \mathbb{R}^n : V_{21}^{\,*} < 1\right\}$) and the magenta dotted line ($\Omega_{22}^{\,*} := \left\{x \in \mathbb{R}^n : V_{22}^{\,*} < 1\right\}$). In the interest of brevity, further shifts are omitted. Finally, the union of the level set is computed using Eq. (22) as follows

$$\begin{gathered} \Omega_e := \{x \in \mathbb{R}^n \mid R_e > 0\}, \\ R_e = R_{\cup}(R_{\cup}(R_{\cup}(R_{\cup}(1 - V_0^{\,*}, 1 - V_1^{\,*}), 1 - V_2^{\,*}), 1 - V_{11}^{\,*}), 1 - V_{21}^{\,*}), 1 - V_{22}^{\,*}) \end{gathered} \tag{42}$$

and

$$\Omega_e = \Omega_0^{\,*} \cup \Omega_1^{\,*} \cup \Omega_2^{\,*} \cup \Omega_{11}^{\,*} \cup \Omega_{21}^{\,*} \cup \Omega_{22}^{\,*}. \tag{43}$$

The union Ω_e can be visualized in Fig. 7 (a).

From the comparison of the A1, A2, and RcomSSF methods shown in Fig.7(a), it can be concluded that the ROA predictions obtained for A1 and A2 are limited by their convergence, and an increased number of iterations (from 30 to 60) cannot facilitate further expansion. Simultaneously, this limitation is circumvented in the case of the proposed approach RcomSSF, and significant estimation improvements are demonstrated, even after several shifting rounds. The shifting process can be continued further and is limited only by the available time and computational resources.

The effect of the shape matrix M on the algorithm performance is also studied. Fig. 7 (c) shows that the estimation obtained using M_1 expands more along the x_1 axis, and maintains a similar expansion along the x_2 axis because more weight is assigned to x_1 in M_1 based on the geometry knowledge of the ellipse. This highlights the effect of the shape matrix and indicates its selection.

Example 4. Here we consider Hahn's example as follows

$$\begin{cases} \dot{x}_1 = -x_1 + 2x_1^2 x_2 \\ \dot{x}_2 = -x_2 \end{cases}. \tag{44}$$

This system has an asymptotically stable EP at the origin, and its exact ROA is known to be $x_1 x_2 < 1$. The determination of ROA has been extensively studied [6,31,51]. The predetermined shape function is given by

$$p(x) = x^T \begin{bmatrix} 14.47 & 18.55 \\ 18.55 & 26.53 \end{bmatrix} x \tag{45}$$

was used in a previous study [31]. ROA estimation can be increased for higher-degree LFs and a composed LF via pointwise maximum or minimum polynomials [31]. Nevertheless, the predictions are highly conservative. To significantly improve the estimations, the use of multiple-shape functions obtained by rotating the major axis of an ellipse every three degrees is proposed in [31]. Consequently, a series of level sets is obtained. The envelopes of these sets are shown in Fig. 8 (a) for comparison. Even though the estimations are improved compared to the single LF method, the estimation is still boxed inside $|x| < 6$. The results of the invariant set method [51] benchmarked against Hahn's example are plotted in Fig. 8. The results are even more conservative. Overall, these methods are not well-suited because of the unbounded nature of the problem.

A1 and A2, and the proposed RcomSSF algorithms are validated using this example. The methods are initialized with $V_0(x)$ and computed using Eq. (16) for $Q = I$

$$V_0(x) = 0.5x_1^2 + 0.5x_2^2 . \tag{46}$$

For comparison, the initial shape function is set with Eq. (45), and a search is performed for a six-degree LF. As shown in Fig. 8 (a), A1 with the fixed shape function yields better estimations than A2 with the adaptive shape function, indicating that the modification in A2 is not necessarily effective. In addition, both A1 and A2 encounter the same problem of convergence, similar to the previously considered cases. Fig. 8 shows that the proposed RcomSSF outperforms the aforementioned methods. The RcomSSF not only significantly extends the estimations towards the unbounded region, but also predicts the boundary of the stability region more precisely. The ROA estimate is achieved after two rounds of shifting, as shown in Fig. 8 (b). The level set $\Omega_0^* := \{x \in \mathbb{R}^n : V_0^* < 1\}$ obtained using A1 is taken as the parent level set. Considering that the exact boundary $x_1 x_2 < 1$ and Ω_0^* are symmetric about the origin, two centers, $x_1^* = [-4,3]$ and $x_2^* = [4,-3]$, are chosen from Ω_0^*, together with $M = I$ to produce the following shape functions

$$p_1(x) = x_1^2 + x_2^2 - 8x_1 + 6x_2 + 25 \quad \text{and} \quad p_2(x) = x_1^2 + x_2^2 + 8x_1 - 6x_2 + 25 . \tag{47}$$

The two obtained proven level sets are represented by the green lines in Fig. 8 (b) (the green solid line $\Omega_1^* := \{x \in \mathbb{R}^n : V_1^* < 1\}$ and the green dashed line $\Omega_2^* := \{x \in \mathbb{R}^n : V_2^* < 1\}$). Next, for the second round of shifts, the centers $x_{11}^* = [-5,5]$ and $x_{12}^* = [-6,2]$ are selected from Ω_1^*, and symmetrical centers $x_{21}^* = [5,-5]$ and $x_{22}^* = [6,-2]$ are selected from Ω_2^*. The resulting level sets are represented using cyan lines. The second round closely approaches the upper and lower boundaries. Moreover, the proposed routine can be performed for the next

round of shifts for further expansion. Using R-composition, these independent level sets are combined into a single set

$$\begin{gathered}\Omega_e := \{x \in \mathbb{R}^n \mid R_e > 0\}, \\ R_e = R_{\cup}(R_{\cup}(R_{\cup}(R_{\cup}(R_{\cup}(1-V_0^*,1-V_1^*),1-V_2^*),1-V_{11}^*),1-V_{12}^*),1-V_{21}^*),1-V_{22}^*)\end{gathered} \tag{48}$$

and

$$\Omega_e = \Omega_0^* \cup \Omega_1^* \cup \Omega_2^* \cup \Omega_{11}^* \cup \Omega_{12}^* \cup \Omega_{21}^* \cup \Omega_{22}^* \tag{49}$$

as shown in Fig. 8 (a).

Example 5. This example is a three-degree Taylor expansion of the system given in [52].

$$\begin{cases} \dot{x}_1 = x_2 + x_3^2 \\ \dot{x}_2 = x_3 - x_1^2 - x_1(x_1 - 1/6x_1^3) \\ \dot{x}_3 = -x_1 - 2x_2 - x_3 + x_2^3 + 1/10\,(2/3x_3^3 + 2/5x_3^5) \end{cases}. \tag{50}$$

It originates from an asymptotically stable EP (0,0) and three other unstable EPs. The ROA estimation results of EP (0,0) for A1, A2, and the proposed RcomSSF method are shown in Fig. 9 (a) and (b), respectively, and the cross-section for $x_2 = 0$ is shown in Fig. 9 (c); the three-dimensional visualization is shown in Fig. 9 (d). The results show that the RcomSSF gives a larger provable level set by shifting to centers $x_1^* = [0.8, 0, 0]$, $x_2^* = [-0.8, 0, 0.6]$, $x_3^* = [0.2, 0, -0.8]$ and $x_4^* = [0, 0, 1.2]$, which are chosen in the level set obtained by A1. We denote Ω_0^* as the set obtained using A1 and $\Omega_i^* (i = 1, 2, 3, 4)$ as the sets obtained during the shifting procedure. Their union via R-composition using Eq. (22) is given by

$$\begin{gathered}\Omega_e := \{x \in \mathbb{R}^n \mid R_e > 0\}, \\ R_e = R_{\cup}(R_{\cup}(R_{\cup}(R_{\cup}(1-V_0^*,1-V_1^*),1-V_2^*),1-V_3^*),1-V_4^*)\end{gathered} \tag{51}$$

and

$$\Omega_e = \Omega_0^* \cup \Omega_1^* \cup \Omega_2^* \cup \Omega_3^* \cup \Omega_4^* \tag{52}$$

This example demonstrates the effectiveness of RcomSSF for a higher dimensional system.

6 Conclusion

Knowledge of the ROA is crucial for nonlinear system analysis and control design. However, existing methods facilitate conservative estimates, especially for non-symmetric or unbounded ROAs. A cost-effective method (RcomSSF) for ROA estimation of nonlinear polynomial systems that demonstrates superior effectiveness compared to other existing approaches is proposed in this study. This method utilizes shape functions with centers that are shifted iteratively closer to the boundary of the obtained proven subset. SOS programming methods are applied to obtain Lyapunov functions for the shifted shape functions. A composition method for Lyapunov functions, namely R-composition, is used in the proposed RcomSSF to combine the resulting independent level sets into a single level set or single function, which results in a compact and richer-shaped expression.

The proposed method is based on SOS optimization techniques and relies on the efficiency of the SOS solver. One of the well-recognized limitations of these solvers is that their computational complexity increases dramatically with the scale of the problem.

Five examples from the literature, including two- and three-dimensional systems with bounded or unbounded, symmetric, or non-symmetric ROA are used to benchmark the proposed RcomSSF compared to existing methods. The results demonstrate the exceptional performance of RcomSSF. The main advantages of the algorithm are the following: improved estimation even when other algorithms encounter estimation limits owing to convergence or numerical infeasibility, improved estimation precision at the cost of a linear increase in the

Journal Pre-proof

computational burden, instead of a polynomial growth in the case of high-degree LFs, and no prior knowledge of the exact ROA. This is a potential avenue for the effective estimation of the ROA for real-world problems.

Acknowledgments

The authors would like to acknowledge the time and effort devoted by reviewers to improving the quality of this work.

References

[1] Khalil HK. Nonlinear systems. 3rd ed. Prentice Hall; 2002.

[2] Khodadadi L, Samadi B, Khaloozadeh H. Estimation of region of attraction for polynomial nonlinear systems: a numerical method. ISA Trans 2014;53:25–32. https://doi.org/10.1016/j.isatra.2013.08.005.

[3] Topcu U, Packard A, Seiler P. Local stability analysis using simulations and sum-of-squares programming. Automatica 2008;44:2669–75. https://doi.org/10.1016/j.automatica.2008.03.010.

[4] Kellett CM. Classical converse theorems in Lyapunov's second method. Discret Contin Dyn Syst - Ser B 2015;20:2333–60. https://doi.org/10.3934/dcdsb.2015.20.2333.

[5] Topcu U, Packard A. Local stability analysis for uncertain nonlinear systems. IEEE Trans Automat Contr 2009;54:1042–7. https://doi.org/10.1109/TAC.2009.2017157.

[6] Genesio R, Tartaglia M, Vicino A. On the estimation of asymptotic stability regions: state of the art and new proposals. IEEE Trans Automat Contr 1985;30:747–55. https://doi.org/10.1109/TAC.1985.1104057.

[7] Parrilo PA. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology, Pasadena, CA, 2000. https://doi.org/10.7907/2K6Y-CH43.

[8] Tan W, Packard A. Stability region analysis using sum of squares programming. 2006 Am. Control Conf., Minneapolis, MN, USA: IEEE; 2006: 2297–302. https://doi.org/10.1109/ACC.2006.1656562.

[9] Tibken B, Fan Y. Computing the domain of attraction for polynomial systems via BMI optimization method. Proc Am Control Conf 2006;2006:117–22. https://doi.org/10.1109/acc.2006.1655340.

[10] Jarvis-Wloszek Z, Feeley R, Tan W, Sun K, Packard A. Some controls applications of sum of squares programming. 42nd IEEE Int. Conf. Decis. Control (IEEE Cat. No.03CH37475), vol. 5, Maui, HI, USA: IEEE; 2003: 4676–81. https://doi.org/10.1109/CDC.2003.1272309.

[11] Kant N, Mukherjee R, Chowdhury D, Khalil HK. Estimation of the region of attraction of underactuated systems and its enlargement using impulsive inputs. IEEE Trans Robot 2019;35:618–32. https://doi.org/10.1109/TRO.2019.2893599.

Journal Pre-proof

[12] Yuan G, Li Y. Estimation of the regions of attraction for autonomous nonlinear systems. Trans Inst Meas Control 2019;41:97–106. https://doi.org/10.1177/0142331217752799.

[13] Sidorov E, Zacksenhouse M. Lyapunov based estimation of the basin of attraction of Poincare maps with applications to limit cycle walking. Nonlinear Anal Hybrid Syst 2019;33:179–94. https://doi.org/10.1016/j.nahs.2019.03.002.

[14] Cunis T, Condomines J-P, Burlion L. Local stability analysis for large polynomial spline systems. Automatica 2020;113:1–5. https://doi.org/10.1016/j.automatica.2019.108773.

[15] Giesl P, Osborne C, Hafstein S. Automatic determination of connected sublevel sets of CPA Lyapunov functions. SIAM J Appl Dyn Syst 2020;19:1029–56. https://doi.org/10.1137/19M1262528.

[16] Cunis T, Condomines J-P, Burlion L. Sum-of-squares flight control synthesis for deep-stall recovery. J Guid Control Dyn 2020;43:1498–511. https://doi.org/10.2514/1.G004753.

[17] Riah R, Fiacchini M, Alamir M. Iterative method for estimating the robust domains of attraction of non-linear systems: application to cancer chemotherapy model with parametric uncertainties. Eur J Control 2019;47:64–73. https://doi.org/10.1016/j.ejcon.2018.12.002.

[18] Pei B, Xu H, Xue Y. Lyapunov based estimation of flight stability boundary under icing conditions. Math Probl Eng 2017;2017:1–10. https://doi.org/10.1155/2017/6901894.

[19] Peyrl H, Parrilo PA. A theorem of the alternative for sos Lyapunov functions. 2007 46th IEEE Conf. Decis. Control, IEEE; 2007: 1687–92. https://doi.org/10.1109/CDC.2007.4434258.

[20] Chesi G. Domain of attraction: analysis and control via sos programming. vol. 415. London: Springer London; 2011. https://doi.org/10.1007/978-0-85729-959-8.

[21] Meng F, Wang D, Yang P, Xie G, Guo F. Application of sum-of-squares method in estimation of region of attraction for nonlinear polynomial systems. IEEE Access 2020;8:14234–43. https://doi.org/10.1109/ACCESS.2020.2966566.

[22] Izumi S, Somekawa H, Xin X, Yamasaki T. Estimation of regions of attraction of power systems by using sum of squares programming. Electr Eng 2018;100:2205–16. https://doi.org/10.1007/s00202-018-0690-z.

[23] Li D, Ignatyev D, Tsourdos A, Wang Z. Region of attraction analysis for adaptive control of wing rock system. 3rd IFAC Conf. Model. Identif. Control Nonlinear Syst. MICNON 2021, vol. 54, Tokyo, Japan: Elsevier Ltd; 2021: 518–23. https://doi.org/10.1016/j.ifacol.2021.10.407.

[24] Li D, Tsourdos A, Wang Z, Ignatyev D. Nonlinear analysis for wing-rock system with adaptive control. J Guid Control Dyn 2022:1–8. https://doi.org/10.2514/1.g006775.

[25] Hafstein S, Giesl P. Review on computational methods for Lyapunov functions. Discret Contin Dyn Syst - Ser B 2015;20:2291–331. https://doi.org/10.3934/dcdsb.2015.20.2291.

[26] Papachristodoulou A, Anderson J, Valmorbida G, Prajna S, Seiler P, Parrilo PA. Sum of squares optimization toolbox for matlab user's guide 2016.

[27] Prajna S, Papachristodoulou A, Parrilo PA. Introducing sostools: a general purpose sum of squares programming solver. Proc. 41st IEEE Conf. Decis. Control. 2002., vol. 1, Las Vegas, NV, USA: IEEE; 2002: 741–6. https://doi.org/10.1109/CDC.2002.1184594.

[28] Seiler P. SOSOPT: a toolbox for polynomial optimization 2013;0:1–11.

[29] Sturm JF. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. Optim Methods Softw 1999;11:625–53. https://doi.org/10.1080/10556789908805766.

[30] Lofberg J. Pre- and post-processing sum-of-squares programs in practice. IEEE Trans Automat Contr 2009;54:1007–11. https://doi.org/10.1109/TAC.2009.2017144.

Journal Pre-proof

[31] Tan W. Nonlinear control analysis and synthesis using sum-of-squares programming. Berkeley: University of California, 2006.

[32] Bai Y, Wang Y, Svinin M, Magid E, Sun R. Function approximation technique based immersion and invariance control for unknown nonlinear systems. IEEE Control Syst Lett 2020;4:934–9. https://doi.org/10.1109/LCSYS.2020.2997600.

[33] Prajna S, Papachristodoulou A, Seiler P, Parrilo PA. SOSTOOLS and its control applications. Lect. Notes Control Inf. Sci., vol. 312, 2005: 273–92. https://doi.org/10.1007/10997703_14.

[34] Jarvis-Wloszek Z. Lyapunov based analysis and controller synthesis for polynomial systems using sum-of-squares optimization. Berkeley: University of California, 2003.

[35] Chakraborty A, Seiler P, Balas GJ. Susceptibility of f/a-18 flight controllers to the falling-leaf mode: linear analysis. J Guid Control Dyn 2011;34:57–72. https://doi.org/10.2514/1.50674.

[36] Tan W, Packard A. Searching for control Lyapunov functions using sums of squares programming. Allert Conf Commun Control Comput 2004:210–9.

[37] Tan W, Packard A. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. IEEE Trans Automat Contr 2008;53:565–71. https://doi.org/10.1109/TAC.2007.914221.

[38] Chesi G. Estimating the domain of attraction via union of continuous families of Lyapunov estimates. Syst Control Lett 2007;56:326–33. https://doi.org/10.1016/j.sysconle.2006.10.012.

[39] Balestrino A, Caiti A, Crisostomi E, Grammatico S. R-composition of Lyapunov functions. 2009 17th Mediterr. Conf. Control Autom., Thessaloniki, Greece: IEEE; 2009: 126–31. https://doi.org/10.1109/MED.2009.5164527.

[40] Balestrino A, Caiti A, Crisostomi E. Logical composition of Lyapunov functions. Int J Control 2011;84:563–73. https://doi.org/10.1080/00207179.2011.562549.

[41] Chakraborty A, Seiler P, Balas GJ. Susceptibility of f/a-18 flight controllers to the falling-leaf mode: nonlinear analysis. J Guid Control Dyn 2011;34:73–85. https://doi.org/10.2514/1.50675.

[42] Pandita R, Chakraborty A, Seiler P, Balas G. Reachability and region of attraction analysis applied to gtm dynamic flight envelope assessment. AIAA Guid. Navig. Control Conf., Reston, Virigina: AIAA; 2009. https://doi.org/10.2514/6.2009-6258.

[43] Sidoryuk ME, Khrabrov AN. Estimation of regions of attraction of aircraft spin modes. J Aircr 2019;56:205–16. https://doi.org/10.2514/1.C034936.

[44] Chakraborty A, Seiler P, Balas GJ. Nonlinear region of attraction analysis for flight control verification and validation. Control Eng Pract 2011;19:335–45. https://doi.org/10.1016/j.conengprac.2010.12.001.

[45] Jones M, Mohammadi H, Peet MM. Estimating the region of attraction using polynomial optimization: a converse Lyapunov result. 2017 IEEE 56th Annu. Conf. Decis. Control, vol. 2018-Janua, IEEE; 2017: 1796–802. https://doi.org/10.1109/CDC.2017.8263908.

[46] Shawki N, Lazarou G, Isenberg DR. Stability and performance analysis of a payload-manipulating robot without adaptive control. Int J Robot Autom 2020;35:23–34. https://doi.org/10.2316/J.2020.206-0077.

[47] Dorobantu A, Seiler P, Balas G. Nonlinear analysis of adaptive flight control laws. AIAA Guid. Navig. Control Conf., Toronto, Ontario, Canada: AIAA; 2010. https://doi.org/10.2514/6.2010-8043.

[48] Ryalo V, Moudgalya KM. Robustness analysis of uncertain, nonlinear systems. Proc Am Control Conf 2000;5:3106–10. https://doi.org/10.1109/acc.2000.879137.

Journal Pre-proof

[49] Davison EJ, Kurak EM. Computational method for determining quadratic Lyapunov functions for nonlinear systems. Automatica 1970;7:627–36. https://doi.org/10.1016/0005-1098(71)90027-6.

[50] Ratschan S, She Z. Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions. SIAM J Control Optim 2010;48:4377–94. https://doi.org/10.1137/090749955.

[51] Valmorbida G, Anderson J. Region of attraction estimation using invariant sets and rational Lyapunov functions. Automatica 2017;75:37–45. https://doi.org/10.1016/j.automatica.2016.09.003.

[52] Wang S, She Z, Ge SS. Inner-estimating domains of attraction for nonpolynomial systems with polynomial differential inclusions. IEEE Trans Cybern 2020:1–14. https://doi.org/10.1109/TCYB.2020.2987326.

Journal Pre-proof

Figures

(Color should be used online please.)

Fig. 1 Demonstration of shape functions: $p_1(x): M = [1,0;0,1], x^* = [0,0]$; $p_2(x): M = [1,1;0,3], x^* = [0,0]$;

Fig. 2 Conceptual diagram of three rounds of shifts in RcomSSF

Fig. 3 Location of shifting centers

Fig. 4 Green line: the union between two ellipses $\hat{R}_1 \bigcup \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcup}(R_1, R_2) > 0 \right\}$; Red line: the intersection between two ellipses $\hat{R}_1 \bigcap \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcap}(R_1, R_2) > 0 \right\}$.

Fig. 5 ROA estimation for Example 1

(a) ROA estimation for A1 and A2 at 30 and 60 iterations

(b) ROA estimation and level sets of $p(x)$ for A2

(c) Shifting details of RcomSSF

(d) ROA estimation for RcomSSF

Fig. 6 ROA estimation for Example 2

(a) ROA estimation for A1 and A2

(b) ROA estimation for RcomSSF

Fig. 7 ROA estimation for Example 3

(a) Vector field and ROA estimation using A1, A2, and RcomSSF

(b) Effect of shifting centers

(c) Effect of shape matrices

Fig. 8 ROA estimation of Example 4

(a) ROA estimation for A1, A2, RcomSSF, and the methods in [31,51]

Journal Pre-proof

(b) Shifting details of RcomSSF

Fig. 9 ROA estimation for Example 5

(a) ROA estimation for A1 and A2

(b) ROA estimation for RcomSSF

(c) Cross-section for $x_2 = 0$

(d) Shifting details of RcomSSF

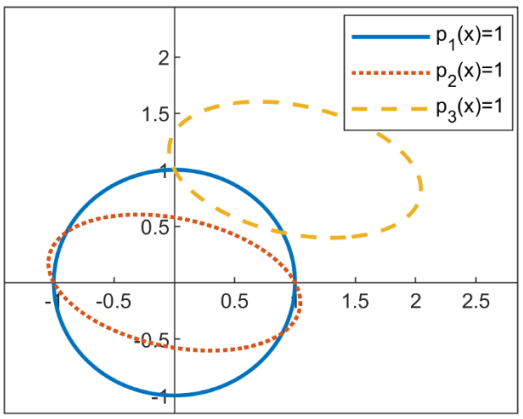


Fig. 1 Demonstration of shape functions: $p_1(x): M = [1,0;0,1],\ x^* = [0,0]$**;**

$p_2(x): M = [1,1;0,3], x^* = [0,0]$**;** $p_3(x): M = [1,0;0,3],\ x^* = [1,1]$

blue solid line: $p_1(x) = 1$; red dotted line: $p_2(x) = 1$; orange dashed line: $p_3(x) = 1$;

Journal Pre-proof

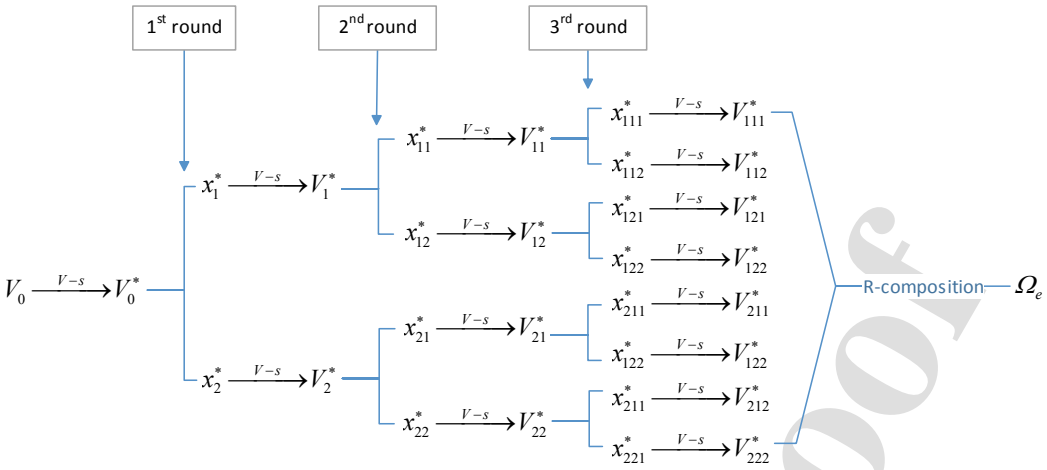


Fig. 2 Conceptual diagram of three rounds of shifts in RcomSSF

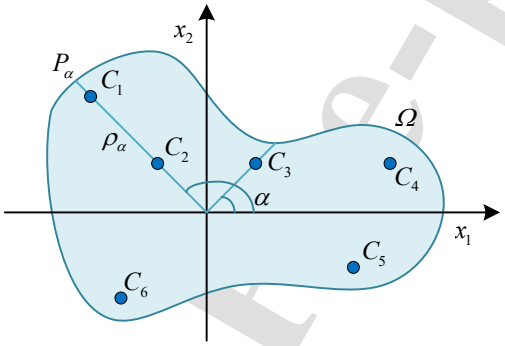


Fig. 3 Location of shifting centers

Journal Pre-proof

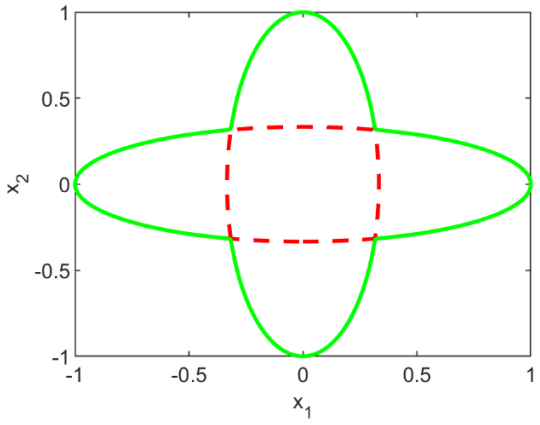
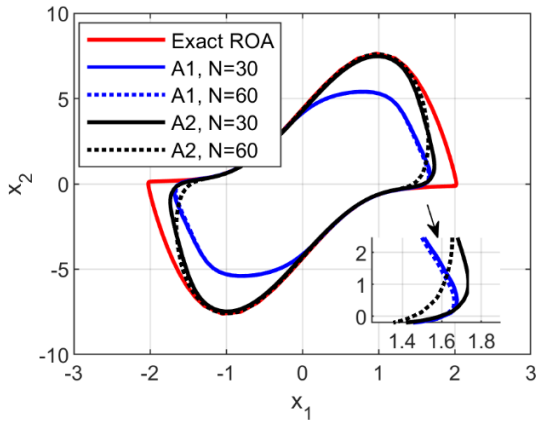
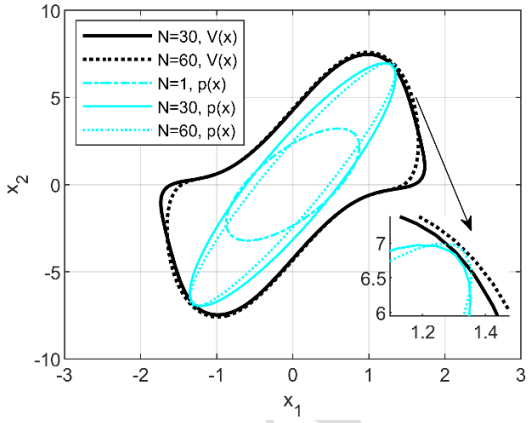


Fig. 4 Green line: the union between two ellipses $\hat{R}_1 \bigcup \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcup}(R_1, R_2) > 0 \right\}$**; Red line: the intersection between two ellipses** $\hat{R}_1 \bigcap \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcap}(R_1, R_2) > 0 \right\}$.

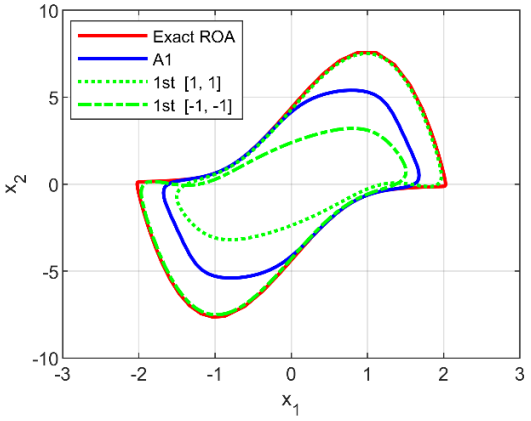
Journal Pre-proof



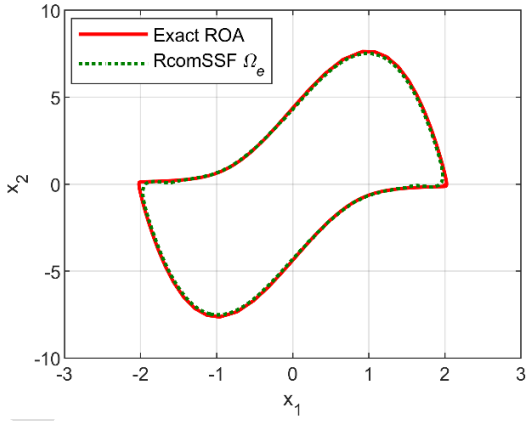
(a) ROA estimation for A1 and A2 at 30 and 60 iterations



(b) ROA estimation and level sets of $p(x)$ for A2

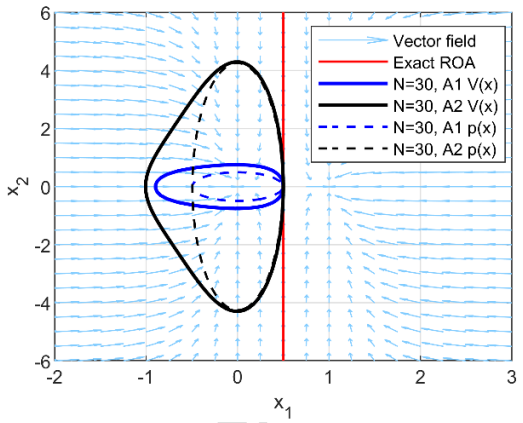


(c) Shifting details of RcomSSF

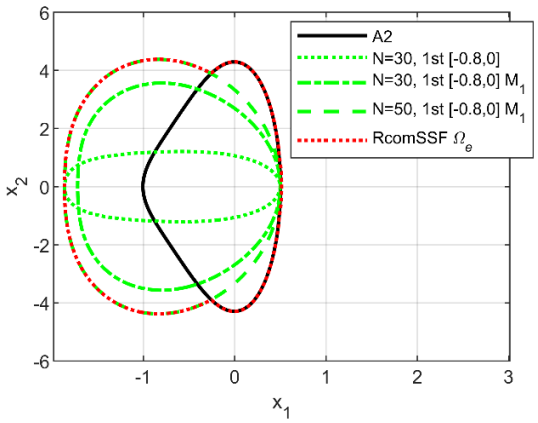


(d) ROA estimation for RcomSSF

Fig. 5 ROA estimation for Example 1

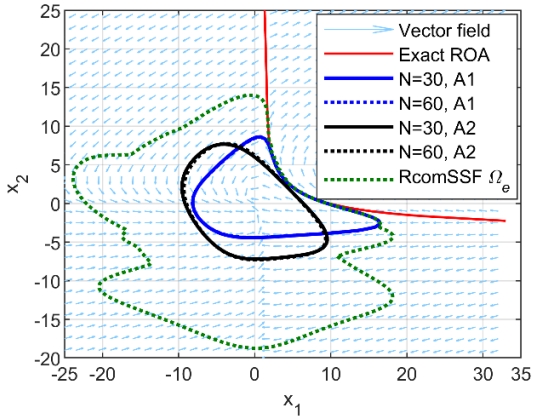


(a) ROA estimation for A1 and A2

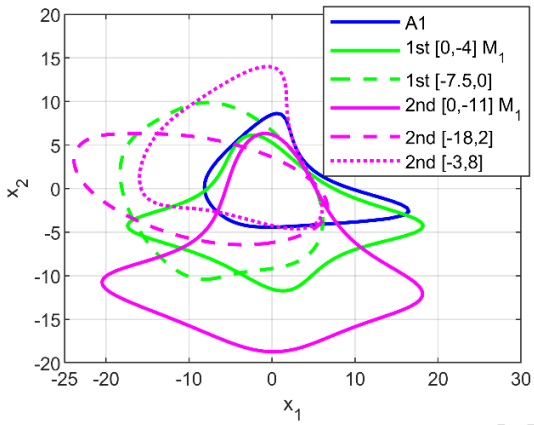


(b) ROA estimation for RcomSSF

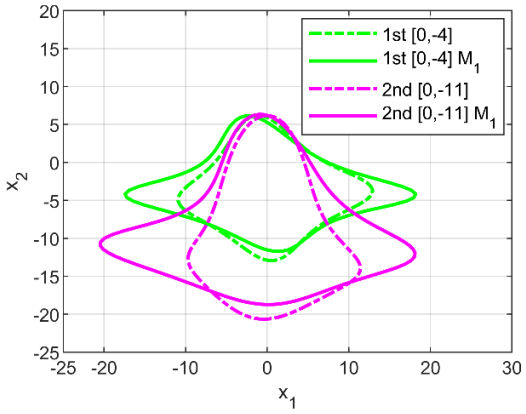
Fig. 6 ROA estimation for Example 2



(a) Vector field and ROA estimation using A1, A2, and RcomSSF

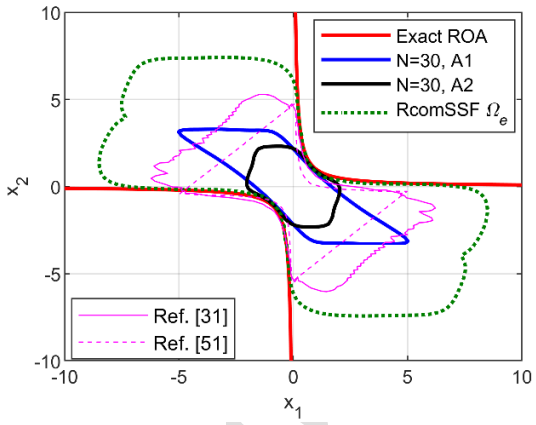


(b) Effect of shifting centers

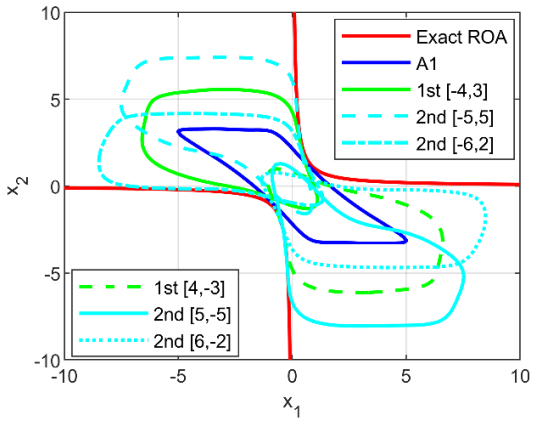


(c) Effect of shape matrices

Fig. 7 ROA estimation for Example 3

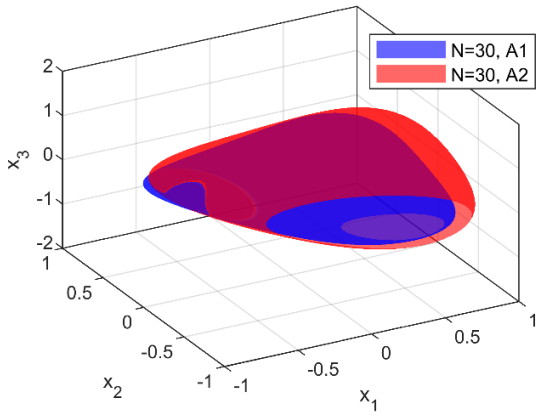


(a) ROA estimation for A1, A2, RcomSSF, and the methods in [31,51]

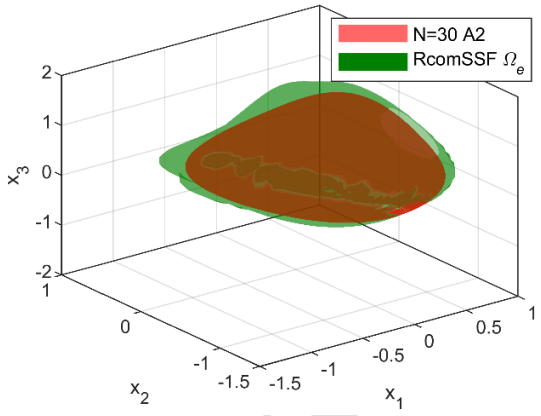


(b) Shifting details of RcomSSF

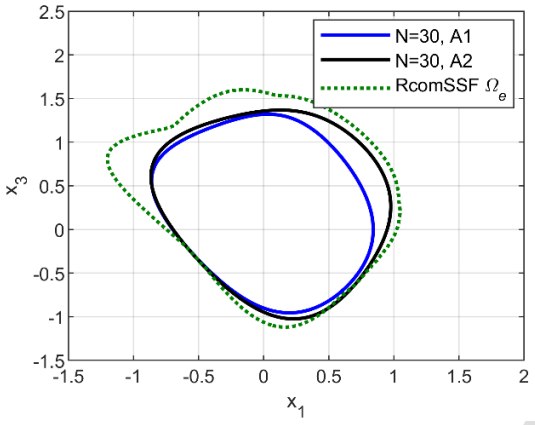
Fig. 8 ROA estimation of Example 4



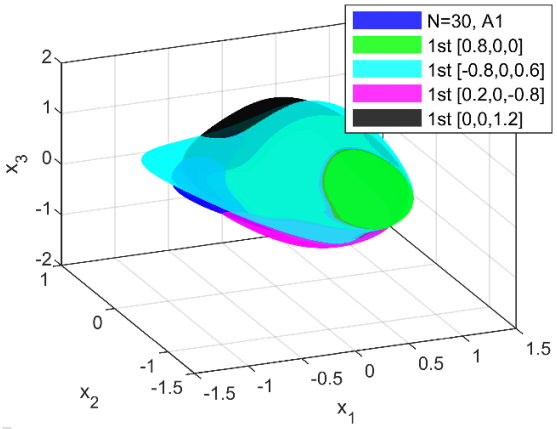
(a) ROA estimation for A1 and A2



(b) ROA estimation for RcomSSF



(c) Cross-section for $x_2 = 0$



(d) Shifting details of RcomSSF

Fig. 9 ROA estimation for Example 5

Journal Pre-proof

Figure legends

Fig. 1, blue solid line: $p_1(x)=1$; red dotted line: $p_2(x)=1$; orange dashed line: $p_3(x)=1$;

Fig. 2, no legend.

Fig. 3, no legend.

Fig. 4, Green line: the union between two ellipses $\hat{R}_1 \bigcup \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcup}(R_1, R_2) > 0 \right\}$; Red line: the intersection between two ellipses $\hat{R}_1 \bigcap \hat{R}_2 = \left\{ x \in \mathbb{R}^n : R_{\bigcap}(R_1, R_2) > 0 \right\}$.

Fig. 5 (a), red sold line: Exact ROA; blue solid line: A1, N=30; blue dotted line: A1, N=60; black solid line: A2, N=30; black dotted line: A2, N=60.

Fig. 5 (b), black solid line: N=30, $V(x)$; black dotted line: N=60, $V(x)$; cyan dash-dot line: N=1, $p(x)$; cyan solid line: N=30, $p(x)$; cyan dotted line: A1, N=60, $p(x)$;

Fig. 5 (c), red sold line: Exact ROA; blue solid line: A1; green dotted line: 1st [1, 1]; green dashed line: 1st [-1, -1]

Fig. 5 (d), red sold line: Exact ROA; green dotted line: RcomSSF Ω_e

Fig. 6 (a), light blue arrow: Vector field; red sold line: Exact ROA; blue solid line: N=30, A1 $V(x)$; black solid line: N=30, A2 $V(x)$; blue dashed line: N=30, A1 $p(x)$; black dashed line: N=30, A2 $p(x)$.

Fig. 6 (b), black solid line: A2; green dotted line: N=30, 1st [-0.8, 0]; green dash-dot line: N=30, 1st [-0.8, 0], M_1; green dashed line: N=50, 1st [-0.8, 0] M_1; red dotted line: RcomSSF Ω_e

Journal Pre-proof

Fig. 7 (a), light blue arrow: Vector field; red sold line: Exact ROA; blue solid line: N=30, A1; blue dotted line: N=60, A1; black solid line: N=30, A2; black dotted line: N=60, A2; green dotted line: RcomSSF Ω_e.

Fig. 7 (b), blue sold line: A1; green solid line: 1st [0, -4] M_1; green dashed line: 1st [-7.5, 0]; magenta solid line: 2nd [0, -11] M_1; magenta dashed line: 2nd [-18, 2]; magenta dotted line: 2nd [-3, 8].

Fig. 7 (c), green dash-dot line: 1st [0, -4]; green solid line: 1st [0, -4] M_1; magenta dash-dot line: 2nd [0, -11] M_1; magenta solid line: 2nd [0, -11] M_1

Fig. 8 (a), red sold line: Exact ROA; blue solid line: N=30, A1; black solid line: N=30, A2; green dotted line: RcomSSF Ω_e; magenta solid line: Ref. [31]; magenta dotted line: Ref. [51]

Fig. 8 (a), red sold line: Exact ROA; blue solid line: A1; green solid line: 1st [-4, 3]; cyan dashed line: 2nd [-5, 5]; cyan dash-dot line: 2nd [-6, 2]; green dashed line: 1st [4, -3]; cyan solid line: 2nd [5, -5]; cyan dotted line: 2nd [6, -2]

Fig. 9 (a), blue: N=30, A1; red: N=30, A2

Fig. 9 (b), red: N=30, A2; green: RcomSSF Ω_e

Fig. 9 (c), blue solid line: N=30, A1; black solid line: N=30, A2; green dotted line: RcomSSF Ω_e;

Fig. 9 (d), blue: N=30, A1; green: 1st [0.8, 0, 0]; cyan: 1st [-0.8, 0, 0.6]; magenta: 1st [0.2, 0, -0.8]; black: 1st [0, 0, 1.2]

Journal Pre-proof

Tables

Table I: Correspondence between logical functions and R-composition

Boolean	Geometry	R-composition
NOT	complement	$-R(x)$
AND	intersection	$R_{\cap}(R_1, R_2) = R_1(x) + R_2(x) - \sqrt{R_1^{\,2}(x) + R_2^{\,2}(x) - \tau R_1(x) R_2(x)}$
OR	union	$R_{\cup}(R_1, R_2) = R_1(x) + R_2(x) + \sqrt{R_1^{\,2}(x) + R_2^{\,2}(x) - \tau R_1(x) R_2(x)}$

Journal Pre-proof

Journal Pre-proof

Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition

Highlights

- A numerical method based on SOS programming is proposed to enlarge the estimation of the region of attraction for locally asymptotically stable equilibrium points of general polynomial systems.
- The method innovatively uses shifted shape functions to enlarge the region of attraction estimation, effective even for non-symmetric or unbounded region of attraction, for which the existing methods present limitations.
- R-composition is employed to express the result in a compact form.
- The proposed method can bring ROA estimation improvement without incurring dramatic computation costs as compared with some existing methods.

Journal Pre-proof

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Journal Pre-proof

https://dspace.org/

Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition

Li, Dongyang; Ignatyev, Dmitry; Tsourdos, Antonios; Wang, Zhongyuan

2022-09-21
Elsevier
https://doi.org/10.1016/j.isatra.2022.11.015
Li D, Ignatyev D, Tsourdos A, Wang Z. (2023) Estimation of non-symmetric and unbounded region of attraction using shifted shape function and R-composition. ISA Transactions, Volume 136, May 2023, pp. 308-322

Downloaded from a DSpace repository. https://dspace.org/