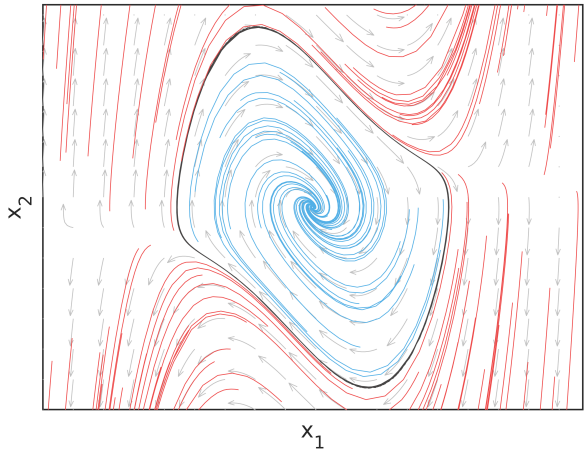
TÉCNICO LISBOA



Analysis and Comparison of Iterative Bilinear Sum-of-Squares Region of Attraction Estimation Algorithms

Renato Bastos Loureiro

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Dr. Torbjørn Cunis
Dr. Rita Cunha

Examination Committee

Chairperson: Dr. Paulo Oliveira
Supervisor: Dr. Torbjørn Cunis
Member of the Committee: Dr. Andrea Iannelli

December 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

“Many places I have been
Many sorrows I have seen
But I don’t forget
Nor will I forget
All who took that road with me”
– Billy Boyd

Acknowledgments

I owe my deepest gratitude and respect to Professor Torbjørn Cunis for all his hard work, availability and patience, it was a pleasure to work alongside you and the knowledge you gave me will for sure help me in my future, so for these reasons, thank you.

I would also like to thank all my friends I made in Stuttgart, it was an incredible experience and I believe it would not have been the same without you, you were patient enough to listen to my endless rants and out of the box ideas even though most of the time you did not understand what exactly I was talking about.

Also, I have to thank all the people I have met along these past five years in Instituto Superior Técnico, including professors who gave me the knowledge and tools to develop this work, and to my beloved friends who I have learn so much and had fun with.

On a more particular note, I would like to thank Dr. Peter Seiler for his advices and opportunity to discuss my ideas with him. Additionally, I would like to thank Rita Cunha, Jan Olucak and Florian Gschwandtner for their availability and feedback throughout this thesis.

Finally, I would like to mention my beloved parents and brothers who have supported me during all these years through some bumpy roads along the way, but also through some really remarkable and memorable moments of my life.

Resumo

Um problema de teoria de sistemas comum para sistemas dinâmicos não lineares críticos, como sistemas aeroespaciais, é a estimativa da região de atração. Uma abordagem bem conhecida é baseada em funções de Lyapunov e conjuntos invariantes. Várias estratégias usando relaxações de soma de quadrados e otimizações semi-algébricas têm sido propostas, mas falta uma comparação dessas estratégias em problemas de referência e uma caracterização detalhada. Propriedades como convergência, robustez e sensibilidade à configuração inicial, questões computacionais e escalabilidade não foram estudadas de forma clara. Nesta tese, é apresentado um conjunto de problemas e procedimentos onde diferentes algoritmos, incluindo um novo algoritmo proposto, devem ser caracterizados, comparados e testados. Uma comparação dos algoritmos mais conhecidos e recentes da literatura é apresentada juntamente com algumas observações sobre implementação e usabilidade. Além disso, um novo método para estimar a região de atração num sistema dinâmico caraterizado no plano com base no teorema de Bendixson-Dulac e relaxações de soma de quadrados é apresentado. Algumas extensões possíveis para dimensões superiores também são apresentadas e comparadas com alguns algoritmos presentes na literatura.

Palavras-chave: Região de Atracção, Sistemas Dinâmicos não Lineares, Polinómios Soma de Quadrados, Programação semi-definida, Função de Lyapunov, Critério de Bendixson-Dulac

Abstract

A common system-theoretic problem for critical nonlinear dynamical systems such as aerospace systems is the estimation of the region of attraction, a well known approach is based on Lyapunov functions and invariant sets. Several strategies using sum-of-squares relaxations and semi-algebraic optimizations have been proposed, but a comparison of those strategies on benchmark problems and a detailed characterization is lacking. Properties such as convergence, robustness to initial guesses, computational issues and scalability have not been studied in a clear way. So in this thesis, a set of problems is presented which build up a proposed test bench where different algorithms are to be characterized, compared and tested, and where all those properties can be extracted. Results from the most known and recent algorithms in the literature are presented as an application of the test bench, alongside a new proposal involving the use of two adaptive shape functions. Additionally, a proposal to estimate the region of attraction through an iterative sum-of-squares approach is presented but in contrast to the current literature, it is not based on Lyapunov theory but instead on the Bendixson-Dulac criterion for planar dynamics and set invariance. Also, extensions for higher dimensional systems are presented and discussed.

Keywords: Region of attraction, Nonlinear dynamical systems, Sum-of-Squares, Semi-definite programming, Lyapunov Function, Bendixson-Dulac Criterion

Contents

List of Tables

List of Figures

Nomenclature

Notation

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Real n-vectors
$\mathbb{R}^{n\times m}$	Real valued matrices with n rows and m columns
$\mathbb{Z}$	Set of integers
$\mathbb{Z}_+^n$	The set of vectors of positive integers of size n
$\mathbb{S}^n$	Set of real symmetric matrices of size n
$\mathbf{M}^\top$	Transpose of $\mathbf{M} \in \mathbb{R}^{n\times m}$
$\mu(\mathbf{A})$	Lozinskii norm of a square matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	Denotes the trace of a square matrix $\mathbf{A}$
$\mathrm{diag}(\cdot)$	Block diagonal matrix of elements in $(\cdot)$
$\mathbb{R}[\mathbf{x}]$	Set of polynomials with real-valued coefficients with variables $\mathbf{x}$
$\mathbb{P}[\mathbf{x}]$	Set of positive semidefinite polynomials with variables $\mathbf{x}$
$\Sigma[\mathbf{x}]$	Set of sum-of-squares polynomials with variables $\mathbf{x}$
$\delta^+(s)$	Maximum degree of polynomial s
$\delta^-(s)$	Minimum degree of polynomial s
∂D	Set of points in the closure of D not belonging to the interior of D
$\mathrm{int}(D)$	Largest open subset of D
$\bar{D}$	Union of D and its boundary ∂D, also called closure of the subset $D \subset \mathbb{R}^n$
$\Sigma M(\mathbb{R}[\mathbf{x}])^2$	Set of polynomial sum-of-squares matrices
$M(\mathbb{R}[\mathbf{x}])$	Set of square matrices with entries belonging to $\mathbb{R}[\mathbf{x}]$
$\mathcal{P}(\mathbb{R}^n)$	Set of subsets of $\mathbb{R}^n$

$L(f,a)$	Denotes the set $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \leq a\}$ with f as a scalar function in $\mathbf{x}$ and $a \in \mathbb{R}$
$L(\mathbf{J},a)$	Denotes the set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{J}(\mathbf{x}) \preceq a\mathbf{I}\}$ with $\mathbf{J}$ as a matrix in $\mathbf{x}$ and $a \in \mathbb{R}$
$\partial L(f,a)$	Denotes the set $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) = a\}$ with f as a scalar function in $\mathbf{x}$ and $a \in \mathbb{R}$
$L^{\circ}(f,a)$	Denotes the set $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < a\}$ with f as a scalar function in $\mathbf{x}$ and $a \in \mathbb{R}$
$L^{\circ}(\mathbf{J},a)$	Denotes the set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{J}(\mathbf{x}) \prec a\mathbf{I}\}$ with $\mathbf{J}$ as a matrix in $\mathbf{x}$ and $a \in \mathbb{R}$

Other symbols

λ	Eigenvalue of a square matrix
$\mathcal{C}^m$	Set of scalar functions which are m-times continuously differentiable
Ω	Region of attraction of a dynamical system
$\otimes$	Kronecker product
$\succ$	A symmetric positive definite matrix $\mathbf{Q}$ is denoted by $\mathbf{Q} \succ 0$
$\succeq$	A symmetric positive semidefinite matrix $\mathbf{Q}$ is denoted by $\mathbf{Q} \succeq 0$
$\mathbf{I}_p$	Identity matrix of size p
$\mathbf{x}_c$	Centroid of a semi-algebraic set
$\mathbf{x}_e$	Equilibrium point

Abbreviations

ROA Region of Attraction.

LF Lyapunov Function.

RTM Reverse Trajectory Method.

BD Bendixson-Dulac.

MLF Maximal Lyapunov Function.

SOS Sum-of-Squares.

P-satz Positivstellensatz.

N-satz Nullstellensatz.

SDP Semi-definite programming.

LMI Linear Matrix Inequality.

BMI Bilinear Matrix Inequality.

CHAPTER 1

Introduction

In the field of dynamic systems and control, one critical aspect the controller design engineer must take into account is stability. Even if a controller is designed to give the system the performance required, it might not meet the requirements for stability, which can lead to issues when used in real-world applications like aerospace systems. Linear analysis is the most common technique for stability analysis and controller development [1], but lacks guarantees and bounds on the stability region. Supplemented with simulation-based techniques, such as Monte Carlo simulations over the entire nonlinear system [2], it provides further information regarding stability bounds and asymptotic stability. Even though this technique seems promising, it is sample-based and does not give an analytical representation on the stability bounds of the system, and actually some nonlinear characteristics can go undetected [3, 4] through the use of this method.

The region of the state space regarding a nonlinear dynamical system where for all initial state vectors within it, the trajectories of the system converge to an equilibrium point $\mathbf{x}_e$ as time goes to infinity is known as the Region of Attraction (ROA) associated with the equilibrium point $\mathbf{x}_e$. The estimation of the ROA has numerous applications such as aerospace systems [5–7], robotics [8], power systems [9–11], chemistry [12–14] and essentially in any system classified as critical.

Computing the exact ROA for nonlinear dynamical systems is very hard if not impossible, only in a few systems in the literature it has been found and typically those systems are simple and have a very well-defined structure.

Numerous methods are put forth in the literature to calculate an approximation of the ROA [15–24]. The dominant approach is based on Lyapunov theory, because it only considers the analytical description of the nonlinear system to certify analytically if a region of the state space is a subset of the true ROA. Even though the Lyapunov theory defines constraints that any Lyapunov Function (LF) must have, they do not give any way or procedure to find either a LF or the ROA. Those constraints are quite difficult to ensure for a general system since they correspond to guarantee positiveness (or negativeness) of functions over a region of the state space.

When dealing solely with polynomial dynamical systems and LFs, the problem of certifying positiveness of functions over a set can be relaxed and efficiently solved.

To certify that a polynomial is positive corresponds to a NP-hard problem but some results from the past century from real algebra, has shown us a way to certify positiveness of polynomials through the use of sum-of-squares relaxations [25, 26]. Employing those results into the Lyapunov's methods and considering a polynomial dynamical system, an estimate of the ROA can be found in a computationally efficient manner by solving iteratively semidefinite programming problems.

Although multiple algorithms have tackled in the last two decades the problem of estimating the ROA through an iterative bilinear sum-of-squares optimization, there is no clear set of properties or well-defined set of problems proposed in the scientific community to study those algorithms in a coherent and well-defined way. Therefore, the purpose of this thesis is to fill the gap in the literature as a set of problems and analysis procedures to not only extract characteristics from each approach but also to compare each one is proposed. Additionally, a new algorithm, corresponding to an enhancement of a well known algorithm, called V-s iteration [27] is presented.

A novel method based on iterative bilinear sum-of-squares algorithms to estimate the ROA on planar dynamics using the Bendixson-Dulac criterion and set invariance is also presented. A few extensions to higher dimensions are also proposed and compared with other algorithms.

Remarks on notation

For ease of notation, given a continuous scalar function $R(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ and a scalar $\gamma \in \mathbb{R}$, we define the set $L(R,\gamma) = \{\mathbf{x} \in \mathbb{R}^n \mid R(\mathbf{x}) \leq \gamma\}$, i.e., $L(R,\gamma)$ is the γ-sublevel set of R. Additionally, we define the set $L^{\circ}(R,\gamma) = \{\mathbf{x} \in \mathbb{R}^n \mid R(\mathbf{x}) < \gamma\}$ for the same R and γ, corresponding to the interior of the γ-sublevel set of R. The γ-level set of R is defined by the set $\partial L(R,\gamma) = \{\mathbf{x} \in \mathbb{R} \mid R(\mathbf{x}) = \gamma\}$. Likewise, given a continuous function $\mathbf{J}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ and a scalar $\gamma \in \mathbb{R}$, we define the sets $L(\mathbf{J},\gamma) = \{\mathbf{x} \in \mathbb{R}^n \mid J(\mathbf{x}) \preceq \gamma\mathbf{I}\}$ and $L^{\circ}(\mathbf{J},\gamma) = \{\mathbf{x} \in \mathbb{R}^n \mid J(\mathbf{x}) \prec \gamma\mathbf{I}\}$.

1.1 Motivation and problem statement

In recent years, applications of sum-of-squares methods for aerospace systems have grown significantly. Multiple methods to estimate the Region of Attraction (ROA) of a nonlinear system based on iterative bilinear sum-of-squares optimization have surged in the literature since the beginning of the 21^{st} century.

The goal of each method is to estimate the ROA described by the semi-algebraic $L(V, \gamma)$ where V is a polynomial in $\mathbf{x}$ and γ is a scalar, that solves the following optimization problem.

Problem 1. *Given* $\mathrm{Vol} : \mathcal{P}(\mathbb{R}^n) \to \mathbb{R}_{\geq 0}$ *and an autonomous dynamical system* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *with* $\mathbf{f} \in \mathcal{C}^1$

$$\begin{aligned} \max_{\mathcal{R}} \quad & \mathrm{Vol}(\mathcal{R}) \\ \text{s.t.} \quad & \mathcal{R} = L(V, \gamma) \subseteq \Omega \end{aligned}$$

where the set Ω *is the true ROA associated with an equilibrium point* $\mathbf{x}_e$*.*

When dealing solely with polynomial systems, Problem 1 can be relaxed such that it is an optimization with a linear cost in one decision variable and a set of constraints that are bilinear in the decision variables.

Several strategies have been proposed in the literature and demonstrated for exemplary systems the capability to solve the Problem 1, yet a comparison or a proposed set of tests does not exist in the scientific community. Each author evaluates and presents their own algorithm on a minor set of examples but a deep analysis into the advantages and drawbacks of it is not clearly shown.

Some current issues in the literature that this work intends to explore and propose solutions are the following.

- There is no defined set of problems and procedures to evaluate and extract properties of each iterative sum-of-squares algorithm that estimates the ROA. Properties such as convergence, robustness to initial guesses, recurrence of computational issues, computational cost and scalability.
- Some important observations in relation to implementation and usability of each proposed method is not stated in the literature.
- The algorithms presented in the literature are tested on a minor set of dynamical systems, most of the time with a well-defined ROA and behavior. There is a lack of information about the performance of the algorithms in chaotic systems and systems with a more complicated dynamic.

As a result, this thesis proposes a unified set of problems to evaluate, validate, and test each algorithm in order to facilitate comparison and make it easier for future papers that propose a new algorithm to estimate the ROA based on iterative bilinear sum-of-squares optimization.

1.2 Contribution

The purpose of this work is to fill the gap in the literature as a set of problems and analysis procedures to not only extract characteristics and properties from each iterative bilinear sum-of-squares ROA estimation algorithm but also to enable comparison between each one is proposed. While working on iterative ROA estimation algorithms, an enhancement of one existent algorithm called V-s iteration, is proposed aiming to tackle some issues of it, such as convergence and accuracy. Additionally, an unconventional strategy for ensuring asymptotic stability, not based on Lyapunov functions, is presented and formulated as an iterative bilinear sum-of-squares algorithm.

The specific contributions of this work are fourfold:

- First, to present an analysis and comparison of some relevant and well known iterative sum-of-squares ROA estimation algorithms.
- Second, to propose a new algorithm, corresponding to an enhancement of one of the algorithms present in the literature in order to tackle some of its issues.
- Third, to inspect the effect of s-multipliers degree choice and propose a rule to choose those based on the constraints and the iterative scheme itself.
- Fourth, to provide a completely different approach to estimate the ROA, not based on Lyapunov theory, but instead on Bendixson-Dulac criterion and on positively invariant sets. A particular strategy for planar dynamics is presented and two algorithm proposals for dynamical systems in any dimension are also presented.

1.3 Thesis Outline

The outline of this thesis is as follows:

Chapter 2 gives the background needed in relation to system dynamics for problem formulation and understanding of the content in the subsequent chapters. Topics such as Lyapunov methods, Nagumo's theorem and set invariance is presented in this chapter.

Chapter 3 gives an overview of the current state of the art for estimating the ROA of nonlinear dynamical systems. Highlighting a few significant traits and characteristics that distinguish each method in the literature.

Chapter 4 provides a brief overview of sum-of-squares polynomials, the Positivstellensatz, semi-definite programming and bilinear sum-of-squares optimization. The aforementioned concepts serve as the foundation for the algorithms developed and discussed throughout this work.

Chapter 5 presents a collection of some pertinent iterative bilinear sum-of-squares ROA estimation algorithms for comparing and testing on a variety of systems throughout this work. In addition, a new algorithm is presented that improves one previously mentioned algorithm while avoiding some shortcomings of the original.

Chapter 6 discusses the problem of selecting the degree of the s-multipliers used in iterative ROA estimation algorithms. Additionally, a suggested rule is offered to select those based solely on the degree of each term involved in the sum-of-squares constraint. The remainder of the thesis employs the method for choosing the s-multiplier degrees described in this chapter.

Chapter 7 studies each algorithm presented in Chapter 5, addressing issues such as accuracy, sensibility to initial guesses, time of computation, numerical issues, feasibility and scalability. Multiple case studies are presented, mostly planar systems but also some significant systems in higher dimensions are considered.

Chapter 8 presents a new iterative sum-of-squares ROA estimation algorithm that it is not based on Lyapunov theory but instead on Bendixson-Dulac criterion and set invariance. The supporting theory and background is presented along with the proposed relaxations and iterative schemes. A discussion is conducted after some results are displayed comparing the proposed algorithms to some Lyapunov-based algorithms.

Chapter 9 presents the conclusions and suggests recommendations for future study in the field of numerical estimation of the ROA of nonlinear dynamical systems using iterative sum-of-squares schemes.

In the **Appendix**, some proofs and system dynamics are presented.

CHAPTER 2

Background

In this chapter, a summary of the background material needed for problem formulation and overall understanding of dynamical systems and stability concepts is presented. A review of Lyapunov's theory and relevant theorems is presented.

2.1 Dynamical systems

Consider the continuous autonomous dynamical system represented by the differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \tag{2.1}$$

where $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is assumed to be globally Lipschitz continuous, and thus, it satisfies sufficient conditions for existence and uniqueness of solutions [28]. The present work only deals with polynomial dynamical systems which are continuous and $\mathcal{C}^\infty$, thus uniqueness of solutions is always satisfied.

Definition 2.1 (Equilibrium point)**.** *Given system (2.1), a point $\mathbf{x}_e$ is an equilibrium point if $\mathbf{f}(\mathbf{x}_e) = 0$.*

Definition 2.2 (Isolated equilibrium point)**.** *A point $\mathbf{x}_e$ is denoted as an isolated equilibrium point of system (2.1) if $\mathbf{x}_e$ is an equilibrium point and there is an $\epsilon > 0$ such that the intersection of the ball centered at $\mathbf{x}_e$ with radius ϵ and the set satisfying $\mathbf{f}(\mathbf{x}) = 0$ only contains $\mathbf{x}_e$, i.e.,*

$$\{\mathbf{x} \in \mathbb{R}^n \mid ||\mathbf{x}_e - \mathbf{x}|| < \epsilon\} \cap \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) = 0\} = \{\mathbf{x}_e\} \tag{2.2}$$

For the remainder of this thesis, whenever we talk about an equilibrium point, we will always be referring to an isolated equilibrium point.

Definition 2.3 (Trajectory)**.** *The function $\Phi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, $\Phi : (\mathbf{x}_0, t) \mapsto \mathbf{x}(t)$ maps a solution to the dynamical system (2.1) with $\mathbf{x}(0) = \mathbf{x}_0$, and this function is called trajectory.*

Definition 2.4 (Orbit)**.** *Given the dynamical system (2.1), the set $\gamma_{\mathbf{x}} = \{\Phi(\mathbf{x}, t) \mid t \geq 0\}$ for any $\mathbf{x} \in \mathbb{R}^n$ is called the orbit through $\mathbf{x}$.*

In some literary works, the word orbit is used in place of the term trajectory, even though by definition, the trajectory is a function which has the orbit as image set, therefore the terms are not entirely interchangeable. For the remainder of this thesis, we will consider orbit as the path or image set of the trajectory, so no confusion arises from the usage of these terms. Additionally, it is relevant to point out that whenever the term trajectory is used we are following the Definition 2.3. Sometimes the term trajectory also accounts for the mapping of solutions over all time and not only non-negative time, as considered here.

There is a particular trajectory in some dynamical systems which is called limit cycle, and it is defined as follows.

Definition 2.5 (Limit cycle [29])**.** *If for any arbitrarily small outer (inner) neighborhood of a given closed trajectory Γ of system (2.1) there exists a non-closed trajectory, then Γ is called external (internal) limit cycle.*

Invariant sets of dynamical systems play an important part in system analysis, as any invariant set bounds the trajectories generated inside the set itself. As it will be shown later, the region of attraction is by definition an invariant set.

Definition 2.6 (Invariant set)**.** *A set $S \subseteq \mathbb{R}^n$ is an invariant set for a dynamical system (2.1) if every point belonging to the trajectory $\Phi(\mathbf{x}_0, t)$ with $\mathbf{x}_0 \in S$ is contained in S, i.e.,*

$$\forall \mathbf{x}_0 \in S, \quad \Phi(\mathbf{x}_0, t) \in S \quad \textit{for} \quad 0 \leq t < \infty \tag{2.3}$$

The definition presented here for invariance of a set is also used to define in the literature positively invariant set or forward invariant set because it accounts for orbits generated for $t \geq 0$. For simplicity, we use the term invariant set with the same meaning as positively invariant set.

In 1942, Nagumo [30] provided necessary and sufficient conditions for invariance of a closed set with respect to system (2.1) by only analyzing the flow on the boundary of the set.

Definition 2.7 (Sub-tangent vector)**.** *A vector $\mathbf{v} \in \mathbb{R}^n$ is sub-tangencial to a set $S \subseteq \mathbb{R}^n$ at $\mathbf{x} \in S$ if*

$$\lim_{\lambda \to 0^+} \inf \frac{\operatorname{dist}(S, \mathbf{x} + \lambda \mathbf{v})}{\lambda} = 0 \tag{2.4}$$

where $\operatorname{dist}(\cdot)$ denotes the Euclidean set distance.

Theorem 2.1 (Nagumo's Theorem [31])**.** *Given system (2.1), let $S \subseteq \mathbb{R}^n$ be a closed set. The set S is positively invariant under the system (2.1) if and only if $\mathbf{f}(\mathbf{x})$ is sub-tangential to S for all $\mathbf{x} \in \partial S$, or equivalently,*

$$\mathbf{f}(\mathbf{x}) \in \mathcal{T}_S(\mathbf{x}), \quad \textit{for all} \quad \mathbf{x} \in \partial S \tag{2.5}$$

where $\mathcal{T}_S(\mathbf{x})$ is the set of all sub-tangencial vectors of S at $\mathbf{x}$.

In particular, given system (2.1), let $h(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ and $h \in \mathcal{C}^1$, consider the set $H = L(h, 0)$ and that $\nabla h(\mathbf{x}) \neq 0$ for all $\mathbf{x}$ satisfying $h(\mathbf{x}) = 0$. Nagumo's theorem gives necessary conditions for set invariance based upon the derivative of h with respect to time on the boundary of H:

$$H \text{ is invariant } \iff \dot{h}(\mathbf{x}) \leq 0 \quad \text{for all} \quad \mathbf{x} \in \partial H$$

This result gives a strategy to obtain invariant sets of system (2.1) by only analyzing the boundary of a compact set characterized by a sublevel set of a single continuously differentiable function.

Definition 2.8. *The Region of Attraction (ROA) of a dynamical system associated with the equilibrium point $\mathbf{x}_e$ is*

$$\Omega = \left\{ \mathbf{x}_0 \in \mathbb{R}^n \mid \lim_{t \to \infty} ||\Phi(\mathbf{x}_0, t) - \mathbf{x}_e|| = 0 \right\} \tag{2.6}$$

Lemma 1. *The ROA of system (2.1) associated with any equilibrium point $\mathbf{x}_e$ is an invariant set.*

Proof. Let $\mathbf{x}_e$ be an equilibrium point of system (2.1). By contradiction, if there exists $\mathbf{x}_0 \in \Omega$ with $\lim_{t\to\infty} ||\Phi(\mathbf{x}_0, t) - \mathbf{x}_e|| = 0$ but $\Phi(\mathbf{x}_0, t) \not\subset \Omega$ for some $t > 0$, due to uniqueness of solution, for any $\mathbf{x}_1 \in \gamma_{\mathbf{x}_0}$, the trajectory generated with that point as initial condition will converge to the equilibrium point, that is, $\lim_{t\to\infty} ||\Phi(\mathbf{x}_1, t) - \mathbf{x}_e|| = 0$. □

Any isolated equilibrium point $\mathbf{x}_e$ can be characterized in the sense of Lyapunov in terms of local stability, depending on the system behavior in its neighborhood.

Definition 2.9 (Local stability of an equilibrium point [32])**.** *The equilibrium point $\mathbf{x}_e$ of (2.1) is*

- *stable if, for each $\epsilon > 0$, there exists a $\delta = \delta(\epsilon) > 0$ such that*

$$\forall t \geq 0, \quad ||\mathbf{x}(0) - \mathbf{x}_e|| < \delta \implies ||\Phi(\mathbf{x}(0), t) - \mathbf{x}_e|| < \epsilon \tag{2.7}$$

- *unstable if it is not stable.*
- *asymptotically stable if and only if it is stable and δ can be chosen such that*

$$||\mathbf{x}(0) - \mathbf{x}_e|| < \delta \implies \lim_{t\to\infty} ||\Phi(\mathbf{x}(0), t) - \mathbf{x}_e|| = 0 \tag{2.8}$$

Definition 2.10 (Global assymptotically stability [32])**.** *An equilibrium point $\mathbf{x}_e$ is globally asymptotically stable if it is stable and for any $\delta \geq 0$, satisfies*

$$||\mathbf{x}(0) - \mathbf{x}_e|| < \delta \implies \lim_{t\to\infty} ||\Phi(\mathbf{x}(0), t) - \mathbf{x}_e|| = 0 \tag{2.9}$$

For simplicity, for the remainder of this thesis whenever it is not clearly stated, the isolated equilibrium point of interest is assumed without loss of generality to be located at the origin ($\mathbf{x}_e = 0$).

2.2 Lyapunov's methods

In Lyapunov's seminal thesis from 1892 presented two key theorems for establishing stability. Those two methods are known as Lyapunov's first method, which states that the local stability of an equilibrium point $\mathbf{x}_e$ can be obtained from studying the linearized system around the equilibrium point. And Lyapunov's second method which characterizes the global stability of the equilibrium point $\mathbf{x}_e$ by using a proper scalar function, called Lyapunov function, defined on the state space.

Theorem 2.2 (Lyapunov's first method [32]). *Let $\mathbf{x}_e$ be an equilibrium point of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ and $\mathbf{J}$ the jacobian matrix of $\mathbf{f}(\mathbf{x})$ at $\mathbf{x}_e$, then:*

1. *$\mathbf{x}_e$ is locally asymptotically stable if all eigenvalues of $\mathbf{J}$ have strictly negative real part;*
2. *$\mathbf{x}_e$ is unstable if any eigenvalue of $\mathbf{J}$ has strictly positive real part.*

Even though Lyapunov's first method is an easy way to check local asymptotic stability, it is only an implication, i.e., an equilibrium point can be locally asymptotically stable but by using the Lyapunov's first method, it is not possible to classify it as locally asymptotically stable.

As an example of the deficiency of using the Lyapunov's first method, consider the polynomial dynamical system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = -\mathbf{x}^3, \quad \text{with} \quad \mathbf{J}(0) = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)(0) = 0$$

It has a local asymptotic stable equilibrium $\mathbf{x}_e = 0$, but it has a null Jacobian matrix at the equilibrium point and by Theorem 2.2, nothing can be said about the equilibrium point.

Lyapunov's second method is a more powerful technique than the first method because it does not rely on the linearization of the system, instead it works directly with the full dynamics of the system.

With the second method, global asymptotic stability can be checked if and only if a function, that is positive everywhere except at the origin, exists such that its derivative is always negative except at the equilibrium point. The corresponding theorem is shown next.

Theorem 2.3 (Lyapunov's second method [32]). *Given system (2.1) with an equilibrium point at $\mathbf{x}_e = 0$, if a continuously differentiable function $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ with $V(0) = 0$ satisfies*

$$\begin{aligned}
&\forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad V(\mathbf{x}) > 0 \\
&\forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad \dot{V}(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) < 0 \\
&V(\mathbf{x}) \to \infty \quad \text{whenever} \quad ||\mathbf{x}|| \to \infty
\end{aligned}$$

then the origin is globally asymptotically stable.

Definition 2.11. *A Lyapunov Function (LF) for system (2.1) with an equilibrium point in $\mathbf{x}_e = 0$ is a scalar strictly positive function $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ with $V(0) = 0$ belonging to $\mathcal{C}^1$ and for which the time derivative $\dot{V}(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$ is non-positive on some region containing the origin.*

Despite Theorem 2.3 being a powerful tool to verify global asymptotic stability, in general non-linear systems have equilibrium points that are not globally asymptotically stable, but instead they have a ROA associated with the equilibrium point which is not the entire space $\mathbb{R}^n$.

Lemma 2, also called Lyapunov's direct method, gives conditions to guarantee that an equilibrium point is locally asymptotically stable, but also provides conditions of a set to be an invariant subset of the ROA associated with the equilibrium point.

Lemma 2 ([32]). *If there exists $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, with $V \in \mathcal{C}^1$ and $V(0) = 0$, a positive scalar γ and let $D = L(V, \gamma) \subseteq \mathbb{R}^n$ with $\mathbf{x}_e = 0$ contained in D, such that*

$$
\begin{aligned}
&D \setminus \{0\} \subseteq L^{\circ}(\nabla V \cdot \mathbf{f}, 0) \\
&D \setminus \{0\} \subseteq L^{\circ}(-V, 0) \\
&D \quad \textit{is bounded}
\end{aligned}
$$

then D is an invariant subset of the ROA associated with the equilibrium point $\mathbf{x}_e = 0$.

Lemma 2 imposes a level set of the LF to be an estimate of the ROA, which is a simple approach, but it is quite conservative. According to LaSalle's theorem [32] it is possible to work instead with a compact set provided that it is positively invariant.

Theorem 2.4 (LaSalle's theorem [32]). *Let $S \subseteq \mathbb{R}^n$ be a compact set that is invariant with respect to system (2.1). Let $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(\mathbf{x}) \leq 0$ in S. Let E be the set of all points $\mathbf{x} \in S$ where $\dot{V}(\mathbf{x}) = 0$ and M be the largest invariant set in E. Then, every solution in S approaches M as $t \to \infty$.*

By Theorem 2.4, if the set M only contains a single equilibrium point, then the set S is a subset of the ROA associated with the equilibrium point.

Lemma 3, presents a set of conditions based on LaSalle's theorem to certify that a set is an invariant subset of the ROA associated with an equilibrium point.

Lemma 3 ([33]). *If there exist R, $V : \mathbb{R}^n \to \mathbb{R}$, with R, $V \in \mathcal{C}^1$, $\gamma \in \mathbb{R}_{>0}$ and let $D = L(R, \gamma) \subseteq \mathbb{R}^n$ with $\mathbf{x}_e = 0$ contained in D such that*

$$
\begin{aligned}
&\partial D \subseteq L^{\circ}(\nabla R \cdot \mathbf{f}, 0) \setminus \{0\} \\
&D \setminus \{0\} \subseteq L^{\circ}(-V, 0) \\
&D \setminus \{0\} \subseteq L^{\circ}(\nabla V \cdot \mathbf{f}, 0) \\
&D \quad \textit{is compact} \\
&V(0) = 0
\end{aligned}
$$

then D is an invariant subset of the ROA associated with the equilibrium point $\mathbf{x}_e = 0$.

CHAPTER 3

State of the Art of ROA Estimation Techniques

The literature offers a variety of methods to estimate the Region of Attraction (ROA) of an isolated equilibrium point of finite-dimensional nonlinear autonomous systems in continuous time characterized by an ordinary differential equation of the following form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \tag{3.1}$$

where $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{f} \in \mathcal{C}^1$ is the vector field. They can be roughly divided into two types of methods depending only on whether they are based on Lyapunov's theory which typically involves optimization problems without the need of trajectory generated data and finding a Lyapunov Function (LF) or not based on Lyapunov's theory which evidently does not require the estimation of a LF of the system. The aforementioned methods can be understood as sample based methods because they often require simulation data to obtain an estimate of the ROA. A more detailed proposed classification of these methods is shown in Figure 3.1.

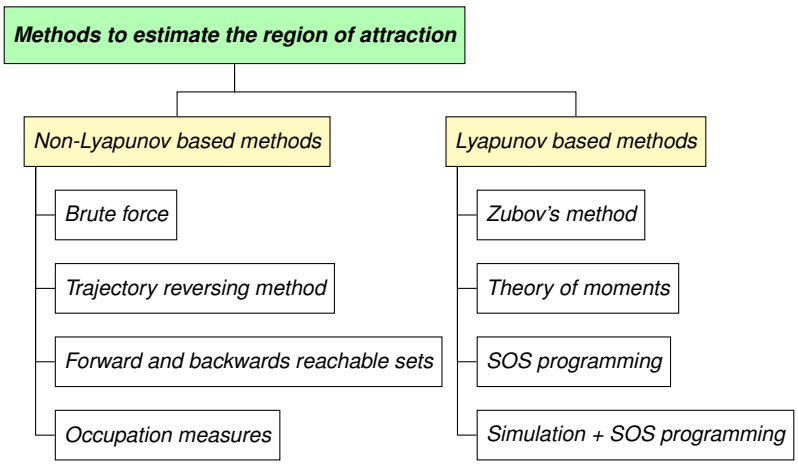


Figure 3.1: Classification of the current nonlinear dynamical systems' ROA estimation algorithms.

The remainder of this chapter provides a general overview of the concepts underlying each method in the literature to estimate the ROA of a nonlinear system. A few remarks on the advantages and disadvantages of each strategy are also included.

3.1 Non-Lyapunov based methods

Non-Lyapunov based methods, as the name suggests, do not use LFs or theory to obtain estimates of the ROA, instead they rely typically on trajectory generation over a finite number of points in the state space to obtain an estimate of it. Even though these methods are not currently predominant in the literature due to having, in general, a high numerical burden, they correspond to the first set of methods developed to estimate the ROA.

The most basic method that is highly inefficient and comes directly from the ROA definition, is the brute force method [34]. This method calls for the discretization of the state space followed by a computation of the trajectory generated by the system dynamics for each point obtained in the discretization. Thus, it is only required to check for each trajectory if it converges to the desired equilibrium point or not.

A technique that is more efficient and provides a better overall estimate of the ROA even for systems with an unbounded ROA, is the Reverse Trajectory Method (RTM) [24, 35]. Instead of checking whether some points are in the ROA by forward simulation, trajectories are generated with the inverse dynamics of the system on points located over a small neighborhood of the equilibrium point which is assumed or guaranteed to be a ROA subset.

Backwards and forward reachable sets techniques are another set of techniques that, when used in practice, produce good results. The computation of the backward reachable sets is done by using the viscosity solution of a time-dependent Hamilton-Jacobi-Isaacs partial differential equation [22] providing an implicit surface representation of the continuous backwards reachable set. As stated in [36], this technique has a few drawbacks, mostly related to computational requirements when the dynamical system dimension grows, that is, the number of states that characterizes a dynamical system increases. As an alternative, a few papers try to estimate the ROA via forward reachable sets. By comparing the two methods, it is shown in [22] that the forward reachable set technique scales better than the backwards technique, which is due to the backwards technique needs for discretization. Even though the forward technique also requires discretization, it has a smaller computational burden in comparison [37].

In contrast to the previous methods, the methods presented in [23, 38] rely on the notion of occupation measures [39] to represent the problem of finding an estimate of the ROA as an infinite-dimensional linear program which can be relaxed to a finite semidefinite program. The advantage of such methods is their overall convexity, meaning that solvers can efficiently solve the inherent optimization.

In general, non-Lyapunov methods do not give an algebraic representation of an estimate of the ROA but only a rough idea of how it is and its shape, also these set of methods have resolution and scalability problems, inherent from the discretization and trajectory generation.

Additionally, most of these methods do not give guarantees over the calculated estimate. The estimate may not be a subset of the ROA, which is problematic because the obtained region can have points that do not belong to the true ROA, and consequently unstable trajectories can be generated from points inside the estimate.

3.2 Lyapunov-based methods

The first conceived method to obtain a LF was described by Zubov in 1964 [18], his method was the first to not only obtain a LF of a dynamical nonlinear system but also allow the construction of the ROA numerically with an algebraic representation. He introduced a first order partial differential equation, called Zubov's equation, of which the solution characterizes the ROA. Despite the fact that theoretically this method provides an exact solution, obtaining an exact algebraic solution is difficult. In general, there is not a purely algebraic technique to solve the partial differential equation, so multiple authors resort to solve the equation numerically [16, 17]. As a result, this requires the use of a grid over the state space where a set of points are used to solve the equation, that is, the equation is solved by a direct discretization of the state space, followed by a minimization problem. This method appears similar to some non-lyapunov methods in that it relies on discretization of the state-space, which as previously stated has resolution and scalability problems. It differs from those methods because there is no need to obtain trajectory data generated by the dynamics of the system in each point of the considered grid.

There are a few workarounds that minimize the effect of discretization. For example, in [15] an adaptive gridding technique is proposed which reduces the sample data required while producing a good solution. In [40], an iterative procedure is proposed which defines the LF as a sum of functions with increasing degree, this method is referred to as Zubov's method. The obtained LF is called Maximal Lyapunov Function (MLF) because it tends to infinity as $\mathbf{x}$ approaches the boundary of the ROA.

In beginning of the 21^{st} century, a huge development in Lyapunov-based methods took place, instead of solving problems based on samples, dynamics-based methods were introduced which did not require discretization of the state space but instead relied on the solution of non-convex problems with polynomial inequalities derived with the Positivstellensatz [41, 42]. By contrast, with the solutions obtained with non-Lyapunov methods, in this set of methods the solution is an estimate of the ROA represented by a semi-algebraic set [43], that is, the subset of the ROA obtained is given by a set of polynomial inequalities.

Some of these methods take as backbone of their architecture Lyapunov's second method [27, 32], which gives conditions to verify global asymptotic stability of a system. Also, on the Lyapunov's direct method which by contrast gives conditions for local asymptotic stability of a Lyapunov level set, resulting in an invariant subset of the ROA of the system. Due to the conservatism [44] of imposing the level sets of a LF to be an estimate of the ROA, LaSalle in [32, 45], suggested that instead of dealing with level sets of a LF, one can just work with compact positively invariant sets, thus evading the conservatism. This allows for more freedom, that is, the function that defines the subset of ROA is not the LF, eventhough the lie derivative of the LF along the dynamics $\mathbf{f}(\mathbf{x})$ needs to be negative over that subset.

The major problem with the conditions established by the Lyapunov's direct method or LaSalle's method is that they impose set containment conditions which are extremely difficult to verify for a general nonlinear dynamical system. For systems with polynomial dynamics, it is possible to verify those conditions through the use of the Positivstellensatz [41, 42], which converts the set containment conditions into a problem of verifying the global positiveness of polynomials. However, determining whether a polynomial is globally positive is an NP-hard problem [46]. A relaxation of the non-negativity condition imposed on the polynomial is suggested in [42] where instead one requires the polynomial to have a sum-of-squares representation. This reduces the challenge to simply finding a positive semi-definite matrix called Gram matrix. As a result it relaxes the original problem into a new one that is P-hard, and consequently it can be solved in polynomial-time. [42]

Regardless of the use of sum-of-squares polynomials, the main problem of finding the biggest subset of the ROA defined by a semi-algebraic set continues to be a non-convex problem [23, 47, 48]. The constraints have bilinear terms featuring the s-multipliers originated from applying the Positivstellensatz to the set containment conditions and other polynomial decision variable functions, such as the LF. Consequently the problem has Bilinear Matrix Inequalities and hence the problem is NP-hard, meaning that there is not an efficient solver to estimate the ROA in polynomial time.

Some works rely on genetic algorithms [24] and particle swarm optimization [49], which are quite efficient methods to solve non-smooth and non-convex optimization problems. However, examples in the literature have been limited to ellipsoids and provided poor results with respect to time of computation and the estimate's accuracy.

One approach to solve the non-convexity of the problem is by dividing the problem into an iterative sequence of optimizations. One can verify that the constraints are bilinear in the decision variables and that the cost function is linear in a single decision variable. It is shown in [27], that this problem can be converted into an iterative sequence of optimizations, where each is quasi-convex [48] and consequently the global optima of each sub-problem can be computed simply by a bisection algorithm. Even though each sub-problem can be solved such that the global optima of each is obtained, the same can not be said about the iterative scheme because it may converge to local optima. Additionally, these schemes require an additional constraint to ensure an enlargement of the estimate of the ROA, that is, it is crucial to add an enlargement constraint to guarantee an increase in the subset of the ROA obtained in each successive iteration.

Some works are presented in [7, 27, 33, 44, 50], each one of these differ from the way they build the sequence of steps and the way they ensure the enlargement of the ROA estimate. In [7, 27], to ensure the enlargement of the estimate, the estimate is constrained by an ellipsoid. This ellipsoid in some works is constant over the optimizations being set at the beginning, but as shown in [50] it reveals some drawbacks, noting that the choice of ellipsoid has an effect on the resulting estimate. To avoid this effect, in [50] an adaptive procedure to change the ellipsoid shape over each iteration is presented, revealing in most cases a better result. Nevertheless, instead of an ellipsoid the containment of the estimate by the previous estimate is possible as shown in [33].

Another method, less common in the literature, is based on the mathematical theory of moments [51] to solve the optimization problem to obtain an estimate of the ROA. An algorithm using this approach is shown in [52]. Just as a remark, this method does not offer an adaptive LF, instead a LF needs to be provided, that is, this method is quite poor because it does not have an adaptive scheme to update the LF.

The techniques mentioned above have an issue, they only handle polynomial systems. Even though, any system can be represented approximately by a polynomial through the use of the Taylor expansion over an equilibrium point, it is underwhelming that these methods only work on these systems. Some recent works, such as the one presented in [53] deals with rational systems instead of solely polynomial systems. Unlike the sum-of-squares approach, the conditions required for the LF are formulated not in a semi-algebraic set but on a compact polytopic set, allowing indefinite matrix solutions for the quadratic decomposition. Other works similar to this one can be found in [54–56].

CHAPTER 4

Sum-of-Squares Programming

In this chapter, the basic concepts behind sum-of-squares relaxations is introduced and their connection to set containment conditions and to iterative bilinear sum-of-squares ROA estimation algorithms is also presented. Concepts such as sum-of-squares polynomial, semi-algebraic set, semi-definite programming and generalized sum-of-squares programming is introduced.

4.1 Sum-of-squares polynomials

Definition 4.1. *(Monomials) A monomial z in n variables is a function defined as $z_\alpha(\mathbf{x}) = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ for $\alpha \in \mathbb{Z}_+^n$. The degree of a monomial is defined as $\sum_{i=1}^n \alpha_i$.*

Definition 4.2. *(Polynomial) A polynomial p in $\mathbf{x} = [x_1, \dots, x_n]^\top$ is a finite linear combination of monomials:*

$$p = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} x_1^{\alpha_1} \dots x_n^{\alpha_n}, \quad c_\alpha \in \mathbb{R} \tag{4.1}$$

where the sum is over a finite number of n-tuples $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$, $\alpha_i \in \mathbb{N}_0$.

The set of all polynomials in $\mathbf{x} = [x_1, \dots, x_n]^\top$ with real coefficients is denoted as $\mathbb{R}[\mathbf{x}]$. The highest degree of a polynomial $p \in \mathbb{R}[\mathbf{x}]$ is defined as the highest degree of its monomials, and it is denoted as $\delta^+(p)$ and equivalently the lowest degree of a polynomial p is defined as the lowest degree of its monomials, and it is denoted as $\delta^-(p)$.

The question of checking whether a polynomial $f \in \mathbb{R}[\mathbf{x}]$ is non-negative or not, i.e., for all $\mathbf{x} \in \mathbb{R}^n$ does $f(\mathbf{x}) \geq 0$ holds is a fundamental mathematical problem in real algebra. As shown in [42], the problem of checking if a polynomial is globally non-negative is NP-hard, thus any algorithm that attempts to solve this problem will certainly have unacceptable behavior from the computational perspective.

In the early two thousands, the introduction of sum-of-squares polynomials enabled to overcome at least comparatively the drawbacks of the problem, solving instead a relaxed version of the original problem. Rather than checking if a polynomial is non-negative, we check if the polynomial is a sum-squares polynomial.

Definition 4.3 (Sum-of-Squares Polynomial)**.** *A multivariate polynomial $p \in \mathbb{R}[\mathbf{x}]$ is a Sum-of-Squares (SOS) polynomial if there exist some polynomials $f_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}],\ i = 1, \ldots, M$ such that*

$$p(\mathbf{x}) = \sum_{i=1}^{M} f_i^2(\mathbf{x}) \tag{4.2}$$

Definition 4.4. *A symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is positive semi-definite (definite) if and only if $\mathbf{x}^\top \mathbf{M}\mathbf{x} \geq 0$ ($\mathbf{x}^\top \mathbf{M}\mathbf{x} > 0$) for all $\mathbf{x} \in \mathbb{R}^n$ or, equivalently, all eigenvalues of $\mathbf{M}$ are non-negative (positive).*

Any polynomial $p \in \mathbb{R}[\mathbf{x}]$ of degree $2k$ that is a SOS polynomial can be represented as

$$p = \mathbf{z}^\top \mathbf{Q}\mathbf{z} \tag{4.3}$$

where matrix $\mathbf{Q}$ is constant positive semi-definite of size l and the vector $\mathbf{z}$ contains monomials of $\mathbf{x}$ of degrees k or lower. The matrix $\mathbf{Q}$ is called *Gram Matrix* and in general for any SOS polynomial there is no unique *Gram Matrix*, this is due to the fact that the vector $\mathbf{z}$ is not composed of independent terms. For ease of notation, the set of SOS polynomials is denoted by $\Sigma[\mathbf{x}]$ and the set of all positive polynomials by $\mathbb{P}[\mathbf{x}]$.

Even though it seems that the problem of polynomial positiveness is solved from the prior representation as a SOS polynomial, this is actually untrue. As proved by Hilbert, actually this class of polynomials $\Sigma[\mathbf{x}]$ only represent a subset of $\mathbb{P}[\mathbf{x}]$ and so not all positive polynomials have a SOS representation. A simple counterexample to the equivalence of SOS polynomials and positive polynomials is the Motzkin polynomial [57], that is as follows,

$$M(x, y, z) \ = \ x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2 \tag{4.4}$$

This polynomial can be easily checked to be a positive polynomial, but it does not have any SOS representation.

Theorem 4.1. *([57]) Let $\mathbb{P}[\mathbf{x}]$ and $\Sigma[\mathbf{x}]$ be defined as above. Then $\Sigma[\mathbf{x}] \subseteq \mathbb{P}[\mathbf{x}]$, but equality holds only for the following cases:*

- *Bivariate forms: number of variables is only 2*
- *Quadratic forms: the degree of the polynomials is 2*
- *Ternary quartics: when the number of variables is 3 and the degree of the polynomial is 4*

With this representation, checking if a polynomial is positive can be relaxed to checking if it is a SOS polynomial. Working solely with SOS polynomials is beneficial because it reduces the problem into finding a positive semi-definite matrix [58], thus converting the original problem into a P-hard problem [42].

4.2 Positivstellensatz

The conditions necessary for a set to be an estimate of the ROA are given as set containment conditions. In this section, some basic concepts of real algebra are presented in order to convert the set containment conditions into a SOS condition.

Definition 4.5 (Semi-algebraic set [43]). *A set $S \subseteq \mathbb{R}^n$ is called semi-algebraic, if S is a Boolean combination (using intersection, unions and complements) of sets of the form*

$$U(f) \;=\; \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \geq 0\}, \tag{4.5}$$

with $f \in \mathbb{R}[\mathbf{x}]$.

Some properties of semi-algebraic sets are the following:

1. $U(1) \;=\; \mathbb{R}^n$ and $U(0) = \emptyset$
2. $\mathbb{R}^n \setminus U(f) \;=\; \{\mathbf{x} \in \mathbb{R}^n \mid -f(\mathbf{x}) \geq 0\}$
3. $\left[\mathbb{R}^n \setminus U(f)\right] \cap \left[\mathbb{R}^n \setminus U(-f)\right] \;=\; \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) = 0\}$

Up until now we discussed how a polynomial can be positive semi-definite over all its domain, and we checked that if it has a SOS representation, it is clearly a non-negative polynomial. Now, one major question which is related to the prior problem about positive polynomials is to check whether a polynomial is non-negative in a semi-algebraic set.

Definition 4.6 (Multiplicative Monoid). *Given $\{g_1, \ldots, g_t\} \in \mathbb{R}[\mathbf{x}]$, the multiplicative Monoid generated by g_j's is the set of all finite products of g_j's, including 1, i.e.,*

$$\mathcal{M}(g_1, \ldots, g_t) = \{g_1^{k_1} \ldots g_t^{k_t} \mid k_1, \ldots, k_t \in \mathbb{Z}_+\}$$

Definition 4.7 (Cone). *Given $\{f_1, \ldots, f_r\} \in \mathbb{R}[\mathbf{x}]$, the Cone generated by f_i's is*

$$\mathcal{P}(f_1, \ldots, f_r) = \left\{ s_0 + \sum_{i=1}^{l} s_i b_i \;\middle|\; l \in \mathbb{Z}_+, s_i \in \Sigma[\mathbf{x}], b_i \in \mathcal{M}(f_1, \ldots, f_r) \right\}$$

Definition 4.8 (Ideal). *Given $\{h_1, \ldots, h_u\} \in \mathbb{R}[\mathbf{x}]$, the Ideal generated by h_k's is*

$$\mathcal{I}(h_1, \ldots, h_u) = \left\{ \sum h_k p_k \mid p_k \in \mathbb{R}[\mathbf{x}] \right\}$$

A central theorem in real algebra that gives an answer to the question of checking non-negativity over a semi-algebraic set is called Positivstellensatz (P-satz), this word is made from the german words *Positiv* (positive), *Stellen* (places) and *Satz* (theorem).

This theorem comes from a specific class of theorems in real algebra called *Stellensätze*. The first theorem from this class, the Nullstellensatz (N-satz), was introduced by Hilbert which dealt with closed fields, such as $\mathbb{C}$, it intends to check whether a set of complex varieties gives a null set or not.

Theorem 4.2 (Nullstellensatz [42]). *Let $\{f_1, \ldots, f_s\}$ be a finite number of polynomials with complex coefficients. The following statements are equivalent:*

1. *The set $\{\mathbf{x} \in \mathbb{C}^n \mid f_i(\mathbf{x}) = 0, \quad i = 1, \ldots, s\}$ is empty.*
2. *There exists polynomials g_i with complex coefficients, such that, $f_1(\mathbf{x})g_1(\mathbf{x}) + \ldots + f_s(\mathbf{x})g_s(\mathbf{x}) = 1$*

Similar to the N-satz, the P-satz is used to check whether a semi-algebraic is empty or not, but it has some crucial differences compared to the original N-satz. The P-satz only concerns with $\mathbb{R}^n$ which is not a closed field.

Theorem 4.3 (P-satz [27]). *Given polynomials $\{f_1, ..., f_r\}$, $\{g_1, ..., g_t\}$ and $\{h_1, ..., h_u\} \in \mathbb{R}[\mathbf{x}]$, the following are equivalent:*

1. *The set below is empty:*

$$\{\mathbf{x} \in \mathbb{R}^n \mid f_1(\mathbf{x}) \geq 0, \ldots, f_r(\mathbf{x}) \geq 0,\ g_1(\mathbf{x}) \neq 0, \ldots, g_t(\mathbf{x}) \neq 0,\ h_1(\mathbf{x}) = 0, \ldots, h_u(\mathbf{x}) = 0\}$$

2. *There exist polynomials $f \in \mathcal{P}(f_1, ..., f_r)$, $g \in \mathcal{M}(g_1, ..., g_t)$, $h \in \mathcal{I}(h_1, ..., h_u)$ such that*

$$f + g^2 + h = 0$$

A relevant example of application of the P-satz theorem involves set containment conditions. The P-satz gives a SOS condition that if it holds the set containment is verified. Lemma 4 displays such application of the P-satz.

Lemma 4 (Simplified P-satz [27]). *Given $h, f_0, \ldots, f_r \in \mathbb{R}[\mathbf{x}]$, the following set containment holds*

$$\{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = 0, f_1(\mathbf{x}) \geq 0 \ldots f_r(\mathbf{x}) \geq 0\} \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid f_0(\mathbf{x}) \geq 0\}$$

if there exist multipliers $p \in \mathbb{R}[\mathbf{x}]$, $s_1, \ldots, s_r \in \Sigma[\mathbf{x}]$ such that

$$p(\mathbf{x})h(\mathbf{x}) - \sum_{i=1}^{r} s_i(\mathbf{x})f_i(\mathbf{x}) + f_0(\mathbf{x}) \in \Sigma[\mathbf{x}]$$

Example 1. *Let $p, q \in \mathbb{R}[\mathbf{x}]$ and $\gamma_1, \gamma_2 \in \mathbb{R}$, if there exist $s_1 \in \Sigma[\mathbf{x}]$ such that for all $\mathbf{x} \in \mathbb{R}^n$, $s_1(p - \gamma_1) + (\gamma_2 - q) \in \Sigma[\mathbf{x}]$ holds, then the set containment condition $L(p, \gamma_1) \subseteq L(q, \gamma_2)$ is satisfied.*

Any constraint of the form $p(\mathbf{x}, \mathbf{a}) \in \Sigma[\mathbf{x}]$ with $\mathbf{x} \in \mathbb{R}^n$ as a free variable and $\mathbf{a} \in \mathbb{R}^m$ as a vector of decision variables, will be denoted as SOS constraint. The variable $\mathbf{a}$ will be omitted when writing SOS constraints when it is evident which polynomial has decision variables.

4.3 Semi-definite programming

Semi-definite programming is a powerful tool in optimization that is used in many contexts, perhaps most notably to obtain strong bounds on discrete optimization problems or non-convex polynomial programs. This section introduces semi-definite programming, the connection between semi-definite programs and SOS polynomials and generalized SOS program.

Definition 4.9 (Linear Matrix Inequality [59])**.** *Let $\boldsymbol{\lambda} \in \mathbb{R}^n$ be a vector of decision variables and symmetric matrices $\{\mathbf{F}_1, \ldots, \mathbf{F}_n\} \in \mathbb{R}^{n \times n}$. A Linear Matrix Inequality is an inequality of the form*

$$\mathbf{F}(\boldsymbol{\lambda}) = \mathbf{F}_0 + \sum_{i=1}^{n} \lambda_i \mathbf{F}_i \succeq 0$$

Definition 4.10 (Semi-definite program)**.** *Let symmetric matrices $\{\mathbf{F}_0, \ldots, \mathbf{F}_n\} \in \mathbb{R}^{n \times n}$ and vector $\mathbf{c} \in \mathbb{R}^n$ be constants and let $\boldsymbol{\lambda} \in \mathbb{R}^n$ be a vector of decision variables. The following optimization problem is called Semi-definite program (SDP),*

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \mathbf{c}^\top \boldsymbol{\lambda} \\ \text{s.t.} \quad & \mathbf{F}(\boldsymbol{\lambda}) = \mathbf{F}_0 + \sum_{i=1}^{n} \mathbf{F}_i \lambda_i \succeq 0 \end{aligned}$$

A Semi-definite programming (SDP) is an optimization problem with a linear cost function subject to a Linear Matrix Inequality (LMI) in the decision variables. One critical aspect of SDPs is their convexity, since their constraints and cost are convex, thus making the optimization problem convex and capable to be solved in polynomial time.

Many optimization problems that appear in engineering systems cannot be written using LMI, instead they can be written in terms of a Bilinear Matrix Inequality, which can be understood as an extension of LMI.

Definition 4.11 (Bilinear Matrix Inequality [59])**.** *Let symmetric matrices $\{\mathbf{G}_1, \ldots, \mathbf{G}_n\}$, $\{\mathbf{H}_{11}, \ldots, \mathbf{H}_{mn}\}$, $\{\mathbf{F}_0, \ldots, \mathbf{F}_m\}$ be constants and let $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ be decision variables. A Bilinear Matrix Inequality (BMI) is of the form:*

$$\mathbf{F}(\mathbf{x}, \mathbf{y}) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i + \sum_{j=1}^{n} y_i \mathbf{G}_j + \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j \mathbf{H}_{ij} \succeq 0 \tag{4.6}$$

A BMI is a linear matrix inequality if either $\mathbf{x}$ or $\mathbf{y}$ is kept fixed. The drawback of BMI is that they are not necessarily convex and consequently, they are more difficult to handle than LMIs [60, 61].

Definition 4.12 (SOS program [62]). *Let $\mathbf{c} \in \mathbb{R}^r$ be a constant and $\mathbf{u} \in \mathbb{R}^r$ be a vector of decision variables. The polynomials $\{a_1, \ldots, a_N\} \in \mathbb{R}[\mathbf{x}]$ are affine in $\mathbf{u}$. An SOS program is an optimization problem of the form,*

$$\begin{aligned} \min_{\mathbf{u}} \quad & \mathbf{c}^\top \mathbf{u} \\ \text{s.t.} \quad & a_k(\mathbf{x}, \mathbf{u}) \in \Sigma[\mathbf{x}], \quad k = 1, \ldots, N \end{aligned}$$

A SOS program is equivalent to a semi-definite program, because a SOS constraint can be interpreted as finding a symmetric matrix $\mathbf{Q}$, affine in the decision variables $\mathbf{u}$, such that $\mathbf{Q} \succeq 0$. Consequently, a SOS program is a convex and efficient solvers can be developed to handle it.

Definition 4.13 (Generalized SOS program [48]). *Let $t \in \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^r$ be decision variables. The polynomials $\{a_1, \ldots, a_N\}$, $\{b_1, \ldots, b_N\}$, $\{c_1, \ldots, c_M\}$ are affine in $\mathbf{u}$. A generalized SOS program is an optimization problem of the form,*

$$\begin{aligned} \min_{t, \mathbf{u}} \quad & t \\ \text{s.t.} \quad & t b_k(\mathbf{x}, \mathbf{u}) - a_k(\mathbf{x}, \mathbf{u}) \in \Sigma[\mathbf{x}], \quad k = 1, \ldots, N \\ & b_k(\mathbf{x}, \mathbf{u}) \in \Sigma[\mathbf{x}], \quad k = 1, \ldots, N \\ & c_k(\mathbf{x}, \mathbf{u}) \in \Sigma[\mathbf{x}], \quad k = 1, \ldots, M \end{aligned}$$

The generalized SOS program is a SOS program where one decision variable is allowed to enter bilinearly in the SOS constraints and therefore, it is non-convex. It is presented in [48], that the generalized SOS program is quasi-convex and consequently, the global minimum can be computed via bisection on t.

The main optimization problem dealt in this thesis is of the following format, where each SOS constraint results from applying the P-satz to a set containment conditions.

$$\begin{aligned} \max_{\gamma, \mathbf{a}, \mathbf{b}} \quad & \gamma \\ \text{s.t.} \quad & -(p_i(\mathbf{x}, \mathbf{a}) - \gamma) s_i(\mathbf{x}, \mathbf{b}) + q_i(\mathbf{x}, \mathbf{a}) \in \Sigma[\mathbf{x}], \quad i = 1, \ldots, M \\ & s_i(\mathbf{x}, \mathbf{b}) \in \Sigma[\mathbf{x}], \quad i = 1, \ldots, M \\ & h_i(\mathbf{x}, \mathbf{a}) \in \Sigma[\mathbf{x}], \quad i = 1, \ldots, N \end{aligned}$$

The terms p_i and s_i are bilinear, making the above problem non-convex. This can be handled by using BMI solvers or by means of iterative SOS schemes. The iterative SOS approach, divides the original problem into a sequence of quasi-convex problems that can be solved efficiently by means of a bisection algorithm, but additional constraints are needed to ensure convergence and enlargement of the set solution.

In the next chapter, a set of iterative bilinear SOS algorithms are presented that estimate a subset of the ROA.

CHAPTER 5

Iterative SOS Algorithms

In order to evaluate and analyze different ROA estimation algorithms based on iterative bilinear sum-of-squares optimization, four algorithms were chosen due to their distinguished iterative procedure, optimization steps and containment conditions involved. Those algorithms are

1. Algorithm from [27] with constant shape function. (Vs)
2. Algorithm from [50] with an adaptive shape function. (Vsp)
3. Invariant set approach (2-steps) from [33]. (IPS)
4. An enhancement of the Vsp algorithm with an additional shape function. (Vsp-c)

Prior sum-of-squares algorithms to estimate the ROA to those mentioned above, such as the one presented in [63], requires a static LF to be given. The invariant subset obtained would be calculated by obtaining the largest sublevel set of the LF such that its derivate with respect to (2.1) would be strictly negative inside the sublevel set, except at the equilibrium point. This is a direct application of Lemma 2. As it is obvious, the solution of this method is highly dependent on the given LF. On the other hand, the advantage is that the problem formulation is quasi-convex, and consequently the global optima would be achieved without resorting to iterative schemes. For the purpose of this thesis, this method will not be explored due to his deficiencies and nonexistence of LF adaptability.

The first three algorithms (Vs, Vsp and IPS) are well known across the literature and differ significantly in how each iterative scheme is carried out. The Vs algorithm was the first to use iterative bilinear sum-of-squares optimization to estimate the ROA while allowing the LF to change over each iteration and avoiding the strict dependence on the initial guess of the LF.

Each algorithm formulation features only bilinear terms, specially with a scalar objective function belonging to those bilinear terms. As shown in [48] if the only bilinearity in the constraints is with a scalar objective function then the problem is quasi-convex and hence the global optima can be obtained via bisection.

A syntactic description of each algorithm is presented next, along with an explanation behind the proposed algorithm, labelled Vsp-c algorithm. Each algorithm description is presented on the left side with the SOS constraints and on the right side with the set containment conditions intended to be guaranteed in each step. To guarantee definite positiveness, an additional function $l(\mathbf{x})$ is added to the SOS constraints.

5.1 V-s iteration

The first iterative sum-of-squares ROA estimation algorithm presented by Tan in [27] is described in Algorithm 1, labelled as Vs algorithm or commonly known as V-s iteration. This approach has some issues, such as being necessary to give an initial guess of the LF, which sometimes it is not a trivial procedure if the system does not admit a local quadratic LF. Typically, the choice of LF is done through the linearized model, but it only works if the resulting linear model is Hurwitz. Also, it is necessary to give a scalar function $p(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ called shape function which commonly is a quadratic function of the form $p(\mathbf{x}) = \mathbf{x}^T\mathbf{N}\mathbf{x}$, with $\mathbf{N} \succ 0$ where each sublevel set is an ellipsoid centered at the origin. The shape function is responsible for the enlargement of the ROA estimate, so a wrong choice leads to poor results as shown in [50]. The shape function should be chosen based on the key direction in the state space, for instance, a common strategy is to use the linearized model and derive a quadratic shape function from it. The disadvantage of this method for obtaining a shape function is that it only considers local behavior, which means that it may lead to subpar results of the estimate of the ROA.

Algorithm 1. ***V-s iteration*** ***(Vs)***

Input: *polynomials V, p and $l = \epsilon(\mathbf{x}^\top\mathbf{x})$ with $\epsilon = 10^{-6}$*

Step I: *hold V and solve for s_1 and γ such that*

$$\begin{aligned} \max_{s_1 \in \Sigma[\mathbf{x}]} \quad & \gamma \\ \text{s.t.} \quad & -(\gamma - V)s_1 - \nabla V \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(V,\gamma) \setminus \{0\} \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \end{aligned}$$

Step II: *hold V and γ and solve for s_2 and β such that*

$$\begin{aligned} \max_{s_2 \in \Sigma[\mathbf{x}]} \quad & \beta \\ \text{s.t.} \quad & (\gamma - V) - s_2(\beta - p) \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \beta \\ \text{s.t.} \quad & L(p,\beta) \subseteq L(V,\gamma) \end{aligned}$$

Step III: *hold s_1, s_2, γ and β and solve for V satisfying*

$$\begin{aligned} & -(\gamma - V)s_1 - \nabla V \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ & (\gamma - V) - s_2(\beta - p) \in \Sigma[\mathbf{x}] \\ & V - l \in \Sigma[\mathbf{x}] \\ & V(0) = 0 \end{aligned} \qquad \begin{aligned} & L(V,\gamma) \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \\ & L(p,\beta) \subseteq L(V,\gamma) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad V(\mathbf{x}) > 0 \\ & V(0) = 0 \end{aligned}$$

Step IV: *Go back to Step I unless the maximum number of iterations is reached or an additional termination condition is met.*

Output: *V and γ*

5.2 V-s iteration with adaptive shape function

Even though the Vs algorithm gives good results, and it is quite simple in its design and implementation, it has the inherent problem of demanding a shape function that will be a static element in the algorithm. To evade the problems of choosing a shape function that may lead to poor results, in [50] an algorithm based on the Vs algorithm is presented that adapts the shape function over each iteration, by replacing the shape function $p(\mathbf{x})$ by the quadratic terms of the LF obtained at the end of each iteration. That algorithm is presented here as Algorithm 2, labelled as Vsp algorithm.

Algorithm 2. ***Vs iteration with p-varying*** ***(Vsp)***

Input: *polynomials V, p and $l = \epsilon(\mathbf{x}^\top\mathbf{x})$ with $\epsilon = 10^{-6}$*

Step I: *hold V and solve for s_2 and γ such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -[(\gamma - V)s_2 + \nabla V \cdot \mathbf{f} + l] \in \Sigma[\mathbf{x}] \\ & s_2 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(V,\gamma) \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \end{aligned}$$

Step II: *hold β and V solve for s_1 and β such that*

$$\begin{aligned} \max \quad & \beta \\ \text{s.t.} \quad & -[(\beta - p)s_1 + (V - \gamma)] \in \Sigma[\mathbf{x}] \\ & s_1 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \beta \\ \text{s.t.} \quad & L(p,\beta) \subseteq L(V,\gamma) \end{aligned}$$

Step III: *hold s_1, s_2, γ and β and solve for V satisfying*

$$\begin{aligned} & -[\nabla V \cdot \mathbf{f} + l + s_2(\gamma - V)] \in \Sigma[\mathbf{x}] \\ & -[(V - \gamma) + s_1(\beta - p)] \in \Sigma[\mathbf{x}] \\ & V - l \in \Sigma[\mathbf{x}] \\ & V(0) = 0 \end{aligned} \qquad \begin{aligned} & L(V,\gamma) \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \\ & L(p,\beta) \subseteq L(V,\gamma) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad V(\mathbf{x}) > 0 \\ & V(0) = 0 \end{aligned}$$

Step IV: *Replace V with V/γ and replace the quadratic part of the new V as new p. Go back to Step I unless maximum number of iterations is reached or an additional termination condition is met.*

Output: *V and γ*

5.3 Positively invariant set

The aforementioned algorithms resort to Lemma 2 in their conceptualization, but as pointed out by LaSalle, choosing the ROA to be a sublevel set of the LF is conservative due to the fact that contractiveness of the level sets defining the ROA is unnecessary. The ROA can be expressed instead in terms of a positively invariant set, i.e., the function that defines the ROA is not necessarily the same as the LF, but the considered set needs to be invariant and every point inside the set must satisfy $\dot{V}(\mathbf{x}) < 0$, for a LF $V(\mathbf{x})$. A few methods based on positively invariant sets and LaSalle's theorem are present in the literature such as in the published works [33, 44].

Algorithm 3, labelled as IPS algorithm presented in [33] uses Lemma 3 and resorts to an enlargement condition based on the constraint of the new estimate of the ROA by the previous estimate, i.e., $L(\bar{R}, \bar{\gamma}) \subseteq L(R, \gamma)$ where $L(\bar{R}, \bar{\gamma})$ denotes the previous estimate and $L(R, \gamma)$ denotes the new estimate.

Algorithm 3. ***Positively invariant set (2 steps)*** ***(IPS)***

Input: *polynomials* R *and* $l = \epsilon(\mathbf{x}^\top \mathbf{x})$ *with* $\epsilon = 10^{-6}$

Step I: *hold* R *and solve for* s_0, s_1, s_2, γ *and* V *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -\nabla R \cdot \mathbf{f} - s_0(\gamma - R) - l \in \Sigma[\mathbf{x}] \\ & V - s_1(\gamma - R) + l \in \Sigma[\mathbf{x}] \\ & -\nabla V \cdot \mathbf{f} - s_2(\gamma - R) - l \in \Sigma[\mathbf{x}] \\ & s_0 \in \mathbb{R}[\mathbf{x}] \\ & s_1,\ s_2 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & \partial L(R, \gamma) \subseteq L^\circ(\nabla R \cdot \mathbf{f}, 0) \\ & L(R, \gamma) \setminus \{0\} \subseteq L^\circ(-V, 0) \\ & L(R, \gamma) \setminus \{0\} \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \end{aligned}$$

Step II: *hold* s_0, s_1, s_2 *and solve for* R, V , γ *and* s_3 *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -\nabla R \cdot \mathbf{f} - s_0(\gamma - R) \in \Sigma[\mathbf{x}] \\ & V - s_1(\gamma - R) \in \Sigma[\mathbf{x}] \\ & -\nabla V \cdot \mathbf{f} - s_2(\gamma - R) \in \Sigma[\mathbf{x}] \\ & (\gamma - R) - s_3(\bar{\gamma} - \bar{R}) \in \Sigma[\mathbf{x}] \\ & s_3 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & \partial L(R, \gamma) \subseteq L(\nabla R \cdot \mathbf{f}, 0) \\ & L(R, \gamma) \setminus \{0\} \subseteq L^\circ(-V, 0) \\ & L(R, \gamma) \setminus \{0\} \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \\ & L(\bar{R}, \bar{\gamma}) \subseteq L(R, \gamma) \end{aligned}$$

Polynomial $\bar{R}$ *and scalar* $\bar{\gamma}$ *correspond to the output of Step I of* R *and* γ*, respectively.*

Step III: *Go back to Step I unless maximum number of iterations is reached or an additional termination condition is met.*

Output: R *and* γ

5.4 V-s iteration with two adaptive shape functions

The new approach presented here is an enhancement of the Vsp algorithm where the ROA estimate is described by a LF sublevel set, and the method adopted to enlarge the estimate is performed through the use of two shape functions. For each of the two shape functions, $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$, the sets given by

$$P_{\beta_1} = L(p_1, \beta_1) \subseteq \Omega$$
$$P_{\beta_2} = L(p_2, \beta_2) \subseteq \Omega$$

with β_1 and β_2 are selected such that the set P_{β_1} and P_{β_2} are the largest sublevel sets of each shape function contained in the prior ROA estimate.

As shown in [50] for the Vs algorithm, the choice of the shape function can affect significantly the shape and size of the ROA estimate, so the choice of the two shape functions, $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$, are based on local behavior around the origin but also on the space distribution of the sublevel set estimate. In the Vsp algorithm, the shape function $p(\mathbf{x})$ is a quadratic function which has an ellipsoid centered at the origin as the shape of its sublevel set. In each iteration the shape function is replaced by the quadratic part of the LF. This shape function update method is intended to improve the ROA estimate over each iteration and avoid the inherent dependence on the initial shape function guess as mentioned before. Some drawbacks or deficiencies arise from using this method, such as oscillations in the estimate volume, i.e., the volume of the estimate does not strictly increase over time. It is observed that this method does not constraint the estimate well when the ROA is not centered at the origin, just like the shape function sublevel set. So employing two shape functions will constrain more the estimate, decreasing the oscillation behavior but also account for non-centered ROA estimates, because one of the shape function's sublevel set is not centered at the origin.

For this purpose, a new algorithm is proposed which uses an additional shape function $p_2(\mathbf{x})$ that seeks to diminish the effect of the prior stated drawbacks. In this case $p_2(\mathbf{x})$ gives a sublevel set which is an ellipsoid that is not necessarily centered at the origin but instead centered at the centroid of the ROA estimate. The format of the shape function $p_2(\mathbf{x})$ sublevel set is determined by the inverse of the covariance matrix of the ROA estimate. The shape function $p_1(\mathbf{x})$ follows the adaptive procedure in the Vsp algorithm.

Consider the function $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, such that

$$g(\mathbf{x}) = \begin{cases} 1, & \text{if} \quad V(\mathbf{x}) \leq \gamma \\ 0, & \text{otherwise} \end{cases}$$

Definition 5.1. *The centroid of a semi-algebraic set is given as*

$$\mathbf{x}_c = \begin{bmatrix} \bar{x}_1 & \cdots & \bar{x}_n \end{bmatrix}^\top = \frac{\int \mathbf{x} g(\mathbf{x}) d\mathbf{x}}{\int g(\mathbf{x}) d\mathbf{x}}$$

Definition 5.2. *The covariance of a semi-algebraic set is*

$$C = \begin{bmatrix} \operatorname{cov}(x_1, x_1) & \cdots & \operatorname{cov}(x_n, x_1) \\ \vdots & \ddots & \vdots \\ \operatorname{cov}(x_1, x_n) & \cdots & \operatorname{cov}(x_n, x_n) \end{bmatrix}, \quad \textit{with} \quad \operatorname{cov}(x_i, x_j) = \frac{\int (x_i - \bar{x}_i)(x_j - \bar{x}_j) g(\mathbf{x}) d\mathbf{x}}{\int g(\mathbf{x}) d\mathbf{x}}$$

The calculation of the centroid and the covariance matrix of the semi-algebraic set that represents the ROA estimate is not an easy problem itself. A few recent works have tried to handle this through an optimization approach [64] but for the purposes of this algorithm, a rough estimate through sampling is enough, and also the time spent in this step is not significant compared to the optimization steps involved. This new proposed algorithm based on the Vsp is described in Algorithm 4, labelled as Vsp-c algorithm.

Algorithm 4. ***Vsp iteration with additional p-inertial*** ***(Vsp-c)***

Input: *polynomials* V, p_1, p_2 *and* $l = \epsilon(\mathbf{x}^\top \mathbf{x})$ *with* $\epsilon = 10^{-6}$

Step I: *hold* V *and solve for* s_2 *and* γ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -[(\gamma - V)s_2 + \nabla V \cdot \mathbf{f} + l] \in \Sigma[\mathbf{x}] \\ & s_2 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(V, \gamma) \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \end{aligned}$$

Step II: *hold* γ *and* V *and solve for* s_1 *and* β_1 *such that*

$$\begin{aligned} \max \quad & \beta_1 \\ \text{s.t.} \quad & -[(\beta_1 - p_1)s_1 + (V - \gamma)] \in \Sigma[\mathbf{x}] \\ & s_1 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \beta_1 \\ \text{s.t.} \quad & L(p_1, \beta_1) \subseteq L(V, \gamma) \end{aligned}$$

Step III: *hold* γ *and* V *and solve for* s_3 *and* β_2 *such that*

$$\begin{aligned} \max_{s_3 \in \Sigma[\mathbf{x}]} \quad & \beta_2 \\ \text{s.t.} \quad & -[(\beta_2 - p_2)s_3 + (V - \gamma)] \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \beta_2 \\ \text{s.t.} \quad & L(p_2, \beta_2) \subseteq L(V, \gamma) \end{aligned}$$

Step IV: *hold* s_1, s_2, s_3, γ, β_1, β_2 *and solve for* V *satisfying*

$$\begin{aligned} & -[\nabla V \cdot \mathbf{f} + l + s_2(\gamma - V)] \in \Sigma[\mathbf{x}] \\ & -[(V - \gamma) + s_1(\beta_1 - p_1)] \in \Sigma[\mathbf{x}] \\ & -[(V - \gamma) + s_3(\beta_2 - p_2)] \in \Sigma[\mathbf{x}] \\ & V - l \in \Sigma[\mathbf{x}] \\ & V(0) = 0 \end{aligned} \qquad \begin{aligned} & L(V, \gamma) \subseteq L^\circ(\nabla V \cdot \mathbf{f}, 0) \\ & L(p_1, \beta_1) \subseteq L(V, \gamma) \\ & L(p_2, \beta_2) \subseteq L(V, \gamma) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad V(\mathbf{x}) > 0 \\ & V(0) = 0 \end{aligned}$$

Step V:

$p_1 \leftarrow$ *quadratic terms of* V.

$p_2 \leftarrow$ *ellipsoid centered at the centroid of the set* $L(V, \gamma)$ *with the Gram matrix equal to the inverse of the covariance matrix of this set.*

Step VI: *Go back to Step I except when maximum number of iterations is reached or an additional termination condition is met.*

Output: V *and* γ

The choice of the new auxiliary shape function $p_2(\mathbf{x})$ makes sense because it is going to be centered at the centroid of the ROA estimate, but it is clear that there are situations where such a choice is going to fail or at least not improve the results. For example, in the case where the ROA estimate is not convex, the centroid of the estimate can be outside the region itself, leading to a zero effect of inserting the new shape function into the algorithm. In other words, the behavior would be identical to the Vsp algorithm.

Figure 5.1 illustrates an example of the application of the Vsp-c algorithm, showing the Lyapunov level set obtained which corresponds to the ROA estimate and the shape functions level set obtained and used to enlarge the estimate.

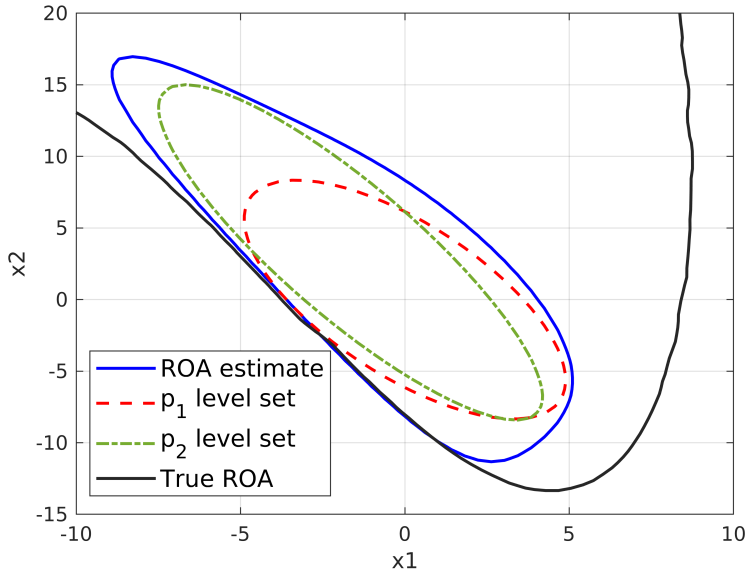


Figure 5.1: Invariant sublevel set of the LF obtained by the Vsp-c algorithm. The level set of both shape functions $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ is also shown.

From Figure 5.1, we observe the constraint done by the sublevel sets of the two shape functions, and that the sublevel set of $p_2(\mathbf{x})$ is not centered at the origin, and it is forcing an enlargement of the ROA subset on the left side.

An evaluation on the performance of this enhancement and a comparison with the other algorithms presented in this chapter is given in Chapter 7.

CHAPTER 6

S-multipliers Issue

In the algorithms presented in Chapter 5, the set of constraints involved are SOS constraints originated from applying the Positivstellensatz (P-satz) shown in Theorem 4.3 to the set containment conditions originated from either Lemma 2 or Lemma 3 but also from the enlargement condition considered in the algorithms. Therefore, according to the P-satz the set containment conditions are verified if it is possible to find values for the decision variables featured in a certain polynomial expression in such a way that that same polynomial is a SOS polynomial. In particular, there are two specific cases frequently used in the formulation of the SOS constraints present in the sum-of-squares ROA estimation algorithms. Those two cases are as follows,

Case 1. *Given* $p,\ q \in \mathbb{R}[\mathbf{x}]$*, if there exist a polynomial* $s \in \Sigma[\mathbf{x}]$ *such that*

$$p \cdot s - q \in \Sigma[\mathbf{x}]$$

holds, then

$$L(p,0) \subseteq L(q,0)$$

is satisfied.

Case 2. *Given* $p,\ q \in \mathbb{R}[\mathbf{x}]$*, if there exist a polynomial* $s \in \mathbb{R}[\mathbf{x}]$ *such that*

$$p \cdot s - q \in \Sigma[\mathbf{x}]$$

holds, then

$$\partial L(p,0) \subseteq L(q,0)$$

is satisfied.

The first case considers a set containment condition described by a sublevel set contained inside another sublevel set. The second case considers a level set contained in a sublevel set.

The polynomial decision variable $s(\mathbf{x})$ originated from using the P-satz is commonly called s-multiplier, and in particular if $s(\mathbf{x})$ corresponds solely to case 1, it is called a SOS multiplier because it is a SOS polynomial.

From a programming perspective, it is necessary to set the minimum and maximum degrees of the s-multipliers involved because it is computationally impossible to work with polynomials of arbitrary degree.

The procedure to select the s-multipliers degree is not straightforward and obvious. If the degree of the s-multipliers is chosen to be below a certain value, the algorithm may not work due to feasibility issues, that is, with the given set of polynomials, it is impossible to get a feasible solution. In contrast, the naive approach of choosing high values for the degrees of the s-multipliers avoids those feasibility issues but increases a lot the computational cost of solving the optimization problem.

For example, as suggested by Tan in [27] in relation to the Vs algorithm, one can just choose the degree of the s-multiplier such that the term with the s-multiplier has at least the same degree as the term with the highest degree in the SOS constraint where that s-multiplier is present. This approach guarantees most of the time a feasible solution, even though sometimes a less restrictive rule such as considering lower degrees of the s-multipliers can give a similar result with a less computational burden due the lower number of decision variables. Even though the procedure suggested by Tan works quite well most of the time, there are situations where it fails when applied to the algorithms presented in Chapter 5.

In the case regarding the enlargement condition employed by an algorithm which imposes that a set $L(\bar{p}, \bar{\gamma})$ with $\bar{p}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ and $\bar{\gamma} \in \mathbb{R}$, corresponding to a previous estimate, is contained inside the set $L(p, \gamma)$ with $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ and $\gamma \in \mathbb{R}$, it has the following corresponding SOS constraints.

$$\begin{aligned} &s_1(\bar{p} - \bar{\gamma}) - (p - \gamma) \in \Sigma[\mathbf{x}] \\ &s_1 \in \Sigma[\mathbf{x}] \end{aligned} \tag{6.1}$$

Some results and experiments with algorithms which used the constraint (6.1) have shown that choosing the s-multiplier $s_1(\mathbf{x})$ to have a degree of zero, stated as the minimum required by Tan's suggestion if both $\bar{p}(\mathbf{x})$ ans $p(\mathbf{x})$ have the same degree, leads to unfeasibility or to a poor result due to computational issues. Similarly, when the SOS constraints are of the form

$$\begin{aligned} &s_2(\bar{p} - \gamma) - q \in \Sigma[\mathbf{x}] \\ &s_2 \in \mathbb{R}[\mathbf{x}] \end{aligned} \tag{6.2}$$

which means that the level set $\partial L(p, \gamma)$ is contained in the sublevel set $L(q, 0)$, we observed through multiple tests on numerous dynamical systems that if the minimum requirements according to Tan are used, the algorithm typically fails. Additionally, we observed that if the s-multiplier $s_2(\mathbf{x})$ is chosen to be an odd degree polynomial, the algorithm frequently fails due to feasibility issues of not choosing the right degree for the s-multiplier.

To show the influence and effect of the s-multipliers $s_i(\mathbf{x})$ on the iterative sum-of-squares ROA estimation algorithms, system (S0) which has exactly one limit cycle bounding the ROA which is hyperbolic and non-algebraic was chosen [65].

$$\dot{\mathbf{x}} = \begin{bmatrix} -x_1 - (x_1 - x_2)(x_1^2 - x_1x_2 + x_2^2) + x_1(2x_1^4 + 2x_1^2x_2^2 + x_2^4) \\ -x_2 - (x_1 + x_2)(2x_1^2 - x_1x_2 + 2x_2^2) + x_2(2x_1^4 + 2x_1^2x_2^2 + x_2^4) \end{bmatrix} \tag{S0}$$

Algorithm 3, labelled as IPS algorithm with constraints similar to the SOS constraints in (6.1) and (6.2) was applied to system (S0) using the setup shown in Table 6.1 considering a single polynomial function of degree four describing the subset of the ROA obtained. The various setups correspond to different sets of s-multipliers' maximum degree considered when applying the algorithm. The minimum degree of each s-multipliers was set to zero.

Table 6.1: Effect of the choice of the degrees of s-multipliers on the estimate of the ROA obtained and on the time of convergence.

setup	$\delta^+(s_0)$	$\delta^+(s_1)$	$\delta^+(s_2)$	$\delta^+(s_3)$	$\mathbf{T}_{iter}$ $[s]$	**Volume**
a	4	2	2	0	6.747	4.903
b	4	2	2	2	7.240	4.905
c	4	2	4	2	6.714	4.908
d	4	2	6	2	7.869	5.314
e	4	4	6	4	6.816	4.909
f	6	4	6	4	9.233	5.317
g	8	8	8	8	13.051	5.318
h	10	10	10	10	22.619	5.318

Some odd and peculiar behaviors are shown in Table 6.1, we expected for the setup **e** to have a better estimate or at least converge to the same results as the setup **d** because it considers higher degree of the s-multipliers, but that was not the case. One possible reason is that the algorithm got into a local optima and was not capable of overcoming the local minima. This undesirable feature is likely due to the iterative scheme, it is shown that each bisection step gives the global optima of the sub-problem, but that does not guarantee that the global optima is going to be reached for the main problem.

Additionally, each bisection step is a sequence of feasibility problems where in each problem, a solution to the constraints is obtained, but it is not the only solution to satisfy the constraints, meaning that the evolution of the algorithm may depend on the choice made by the solver.

Choosing a higher degree of the s-multipliers intuitively avoids feasibility problems and leads in general to better results, but it has the drawback of increasing the time spend per iteration due the increase amount of decision variables as presented in Table 6.1. Also, the estimate evolution over iterations gets slower, that is, the rate in which the volume of the estimate increases over each iteration gets slower for a higher choice of degrees of the s-multipliers, as it can be seen in Figure 6.1 for the setups **d**, **f**, **g** and **h**.

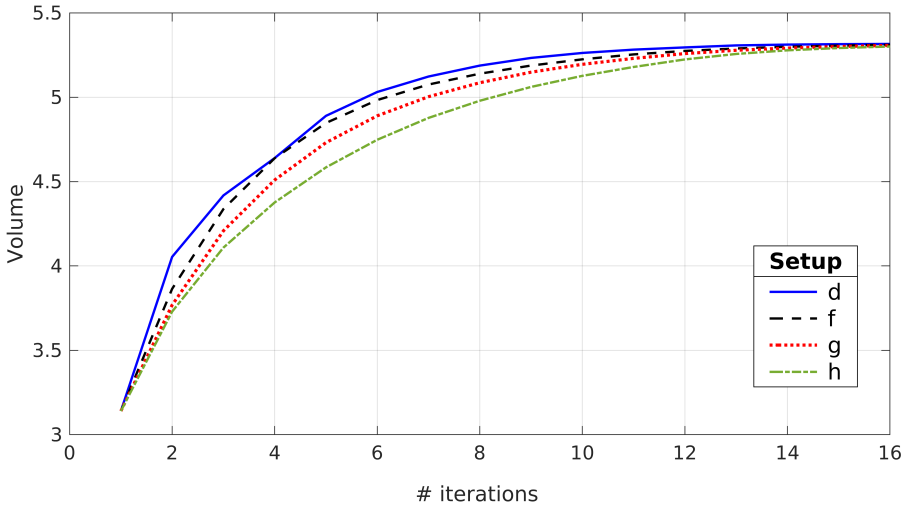


Figure 6.1: Effect of the degree of the s-multipliers on the evolution of the estimate of the ROA over iterations.

According to the aforementioned results and observations, the choice of the s-multipliers degrees is a key factor in the performance, either regarding the achieved estimate's volume or the computation cost in estimating a maximum subset of the ROA. To simplify the s-multipliers degrees choice for each case study and iterative sum-of-squares ROA considered, we propose a procedure to automatically choose those degrees. This policy is purely conceived by means of multiple tests and observations that reveled that the proposed procedure gives good results in general, without a high frequency of numerical problems and also without a high computational burden for all the algorithms considered.

For each multiplier $s_i(\mathbf{x})$ present in a SOS constraint of the form $p_i(\mathbf{x})s_i(\mathbf{x}) + g_i(\mathbf{x}) \in \Sigma[\mathbf{x}]$, the maximum degree of $s_i(\mathbf{x})$ is set to $\delta^+(s_i) = \delta^+(q) - \delta^+(p)$ if the resulting degree is even, or it is set to $\delta^+(s_i) = \delta^+(q) - \delta^+(p) + 1$ otherwise. In case that the resulting degree is equal to zero or less, it is established that the maximum degree is two. The minimum degree is set to $\delta^-(s_i) = 0$. This procedure is illustrated as a decision tree in Figure 6.2.

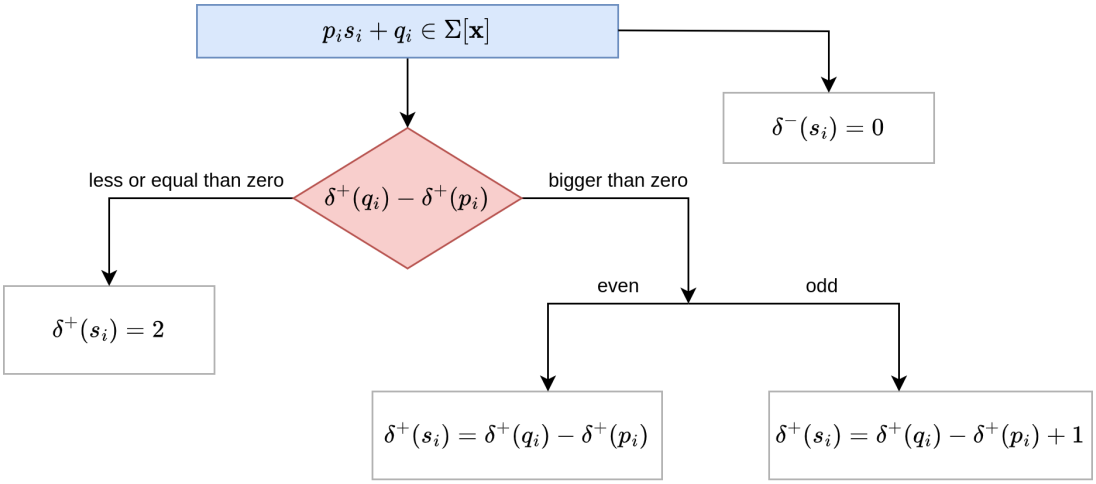


Figure 6.2: Proposed decision tree for S-multiplier selection.

CHAPTER 7

Algorithm Analysis and Comparison

In this chapter, we present multiple case studies and tests to analyze and compare each iterative bilinear sum-of-squares algorithm presented in Chapter 5. Properties such as accuracy, robustness to initial guess, recurrence of numerical issues, speed, behavior over iterations and scalability is considered. All results and analysis were performed on a 2.9Ghz quad-core PC with 8 GB RAM and the chosen SDP solver was SeDuMi [66] alongside SOSOPT library [67]. The choice of the s-multipliers is done by using the procedure presented in Chapter 6.

7.1 Planar systems

Six particular case studies in two dimensions with different characteristics were investigated in order to extract properties and behavior of the algorithms presented in Chapter 5. They differ from each other with regard to the geometry of the true ROA, for example whether the ROA is bounded or unbounded.

Those six polynomial systems are as follows.

$$\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ x_1^2 x_2 - x_1 - x_2 \end{bmatrix} \quad \text{(S1)} \qquad \dot{\mathbf{x}} = \begin{bmatrix} -x_1 + (x_1^2 - 5x_1 + x_2^2)^2 x_1 \\ -x_2 + (x_1^2 - 5x_1 + x_2^2)^2 x_2 \end{bmatrix} \quad \text{(S4)}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 2x_1^2 x_2 - x_1 \\ -x_2 \end{bmatrix} \quad \text{(S2)} \qquad \dot{\mathbf{x}} = \begin{bmatrix} x_1 x_2 - x_1 \\ -x_2 \end{bmatrix} \quad \text{(S5)}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} -x_1 + x_2 \\ -0.1x_1^3 - x_1^2 + 0.1x_1 - 2x_2 \end{bmatrix} \quad \text{(S3)} \qquad \dot{\mathbf{x}} = \begin{bmatrix} x_1^3 - x_1 \\ x_2^3 - x_2 \end{bmatrix} \quad \text{(S6)}$$

The Van der Pol system, defined here by system (S1) is a well known dynamical system in the literature [50, 68, 69]. It is distinguished from other planar dynamical systems because it has a bounded ROA, that it is enclosed by a limit cycle which has no analytical representation. Non-Lyapunov methods such as the Reverse Trajectory Method (RTM) [35] give a really accurate estimate of the ROA of this system without much effort, and it is due to this characteristic that this system is frequent in the literature.

System (S2), also known as Hahn's system [70], has an unbounded ROA that is centered at the origin, that is, it presents a symmetry in relation to the origin. Additionally, this system has an analytic expression for its ROA which consists in two linear polynomial inequalities. The estimation of its ROA has been intensively studied in [27, 35, 44].

System (S3) has an unbounded ROA [63], but unlike system (S2) it is not centered at the origin, meaning that no symmetry exists in the ROA in relation with the origin. This characteristic is interesting because some algorithms rely on the use of ellipsoids centered at the origin, and it is likely that for this system those algorithms may have poor results.

System (S4) has a bounded non-convex ROA, with a shape similar to a boomerang. This particular characteristic is relevant to be tested out by the ROA estimation algorithms because non-convexity of a ROA may constrain or give difficulties to the algorithms.

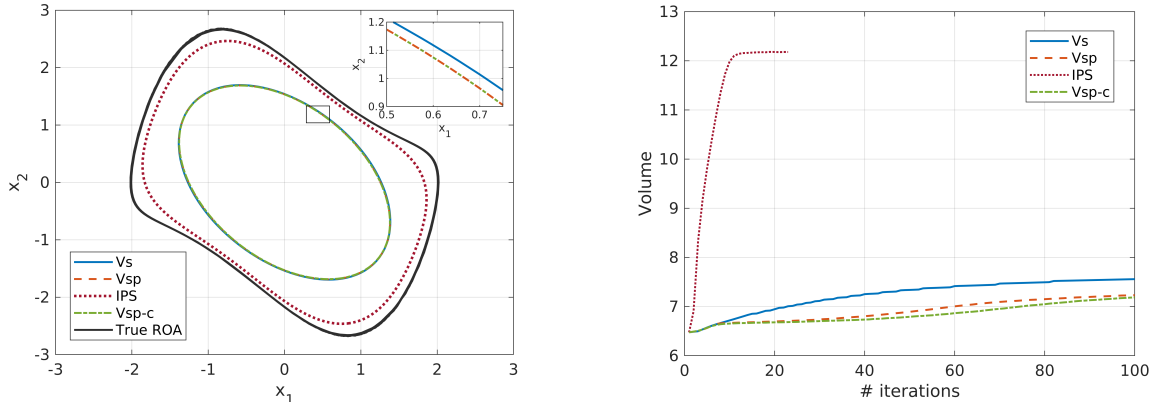
System (S5) is remarkable because unlike the other systems in $\mathbb{R}^2$ presented here, this system is globally asymptotically stable and does not admit a polynomial LF of any degree, as proved in [71]. Taking into consideration that the algorithms worked upon in this thesis rely solely on polynomials, it is interesting to test them out on a system that actually does not have a polynomial LF that describes the whole ROA.

System (S6) has a ROA which is not representable by a single finite degree polynomial sublevel set, because the boundary of the true ROA is not continuously differentiable. This system has an analytical representation of its ROA defined by two linear polynomial inequalities.

The results presented in this chapter originate from applying the algorithms presented in Chapter 5 to the planar dynamical systems described above. Two different graphs are presented for each case, one with the resulted subset of the ROA and another with the subset development over iterations, that is, for each iteration the volume of the new estimate is presented, showcasing the behavior of each algorithm over iterations. With the later graph, some statements about convergence rate and oscillatory behavior of each algorithm can be formed and discussed.

Also, it is important to note that in all the results presented, we considered a ROA subset characterized by a polynomial of degree four, for example in the case of the Vs algorithm this means that the function V has polynomial degree four and in the case of the IPS, means that the function R has polynomial degree four. Additionally, the choice of the s-multipliers degree required by each algorithm is done following the procedure proposed in Chapter 6.

First, let us start our analysis over planar dynamical systems with the all-time favorite planar system in the literature, system (S1). Figures 7.1(a) and 7.1(b) present the results of the ROA subset obtained and its volume evolution over iterations by each algorithm applied to system (S1).



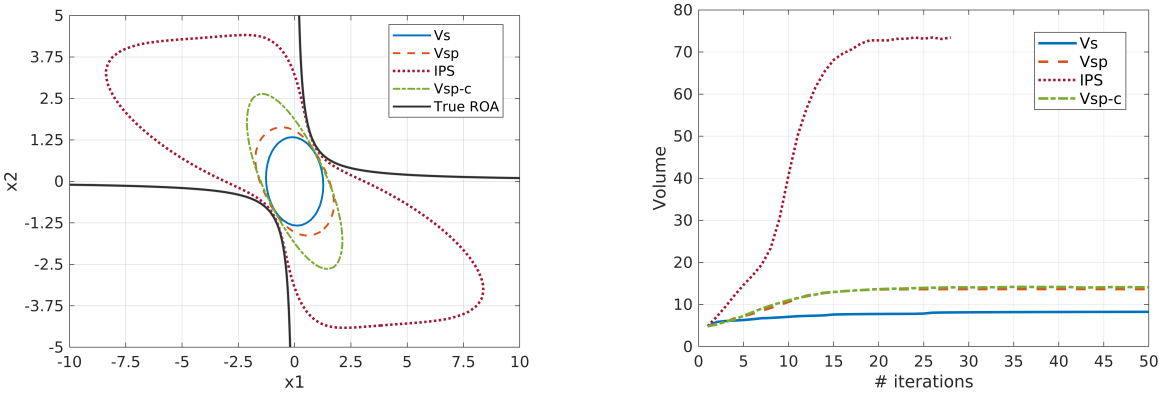
(a) Estimated ROA subset.

(b) Volume evolution over iterations of the ROA subset.

Figure 7.1: Results of algorithm testing for system (S1).

Figure 7.1(a), shows a superior performance of the IPS algorithm in comparison with the rest, showcasing a close ROA estimate to the true ROA. With Figure 7.1(b) we see one unexpected result, the Vs algorithm has a better overall performance and a bigger final estimate volume than the Vsp and Vsp-c algorithms. From a few changes to the s-multipliers, we checked that the reason these odd results appeared is due to the choice of the s-multipliers and also due to the adaptive method that the Vsp and Vsp-c algorithms use to enlarge the ROA subset. It is relevant to point out that the IPS not only gives a better estimate, but also it is the fastest algorithm out of the four presented to converge to its final subset, after 10 iterations the IPS has already converged, as seen in Figure 7.1(b).

Let us now consider system (S2) that has an unbounded ROA centered at the origin, that is, it has a symmetry in relation to the origin. The true ROA of system (S2) is the set $\{\mathbf{x} \in \mathbb{R}^2 \mid x_1 \cdot x_2 < 1\}$ with $\mathbf{x} = [x_1 \quad x_2]^\top$. The results from applying each algorithm to system (S2) are presented in Figures 7.2(a) and 7.2(b).



(a) Estimated ROA subset.

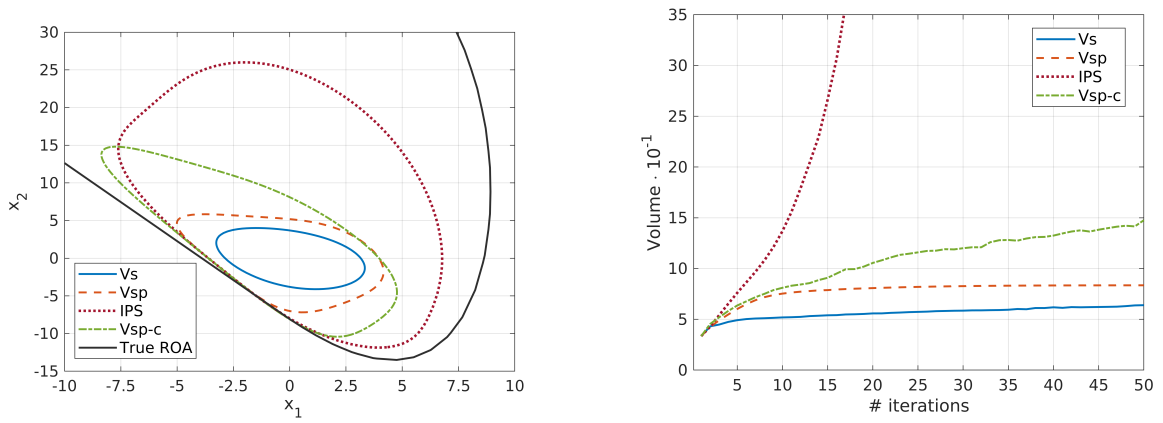
(b) Volume evolution over iterations of the ROA subset.

Figure 7.2: Results of algorithm testing for system (S2).

The results displayed in Figures 7.2(a) and 7.2(b) show that the IPS algorithm gives the best result for the estimate of the ROA because the ROA subset obtained has a volume close to five times the volume of the subsets of the ROA obtained by the other algorithms.

Additionally, the Vsp-c algorithm obtains a slightly better result in comparison to Vsp algorithm, this is due to the fact that the Vsp-c algorithm constrains more the ROA subset leading to a larger enlargement of the subset. In this case due to the symmetry of the system's ROA, there is no big advantage in using the Vsp-c compared to the IPS, because enlarging the ROA through an ellipsoid will not be able to constrain the subset enough for it to be as big as it could.

System (S3) is a good example to test out the algorithms that use a shape function with a sublevel set with the form of an ellipsoid centered at the origin to enlarge the ROA. The ROA of this system is not centered in zero, at least locally, i.e., the ROA has no kind of local symmetry in relation to the origin. It is expected that those algorithms that use shape functions have a poor performance when applied to system (S3). The results from applying the four algorithms to system (S3) are presented in Figures 7.3(a) and 7.3(b).



(a) Estimated ROA subset.

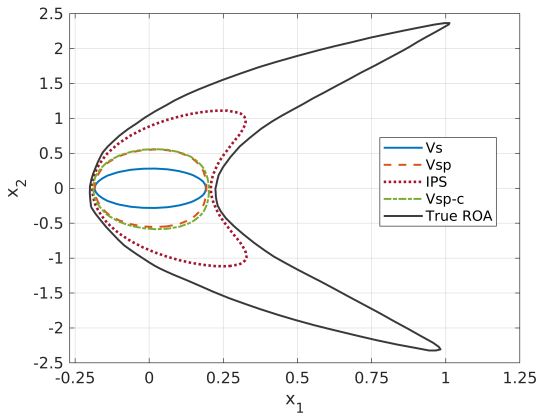
(b) Volume evolution over iterations of the ROA subset.

Figure 7.3: Results of algorithm testing for system (S3).

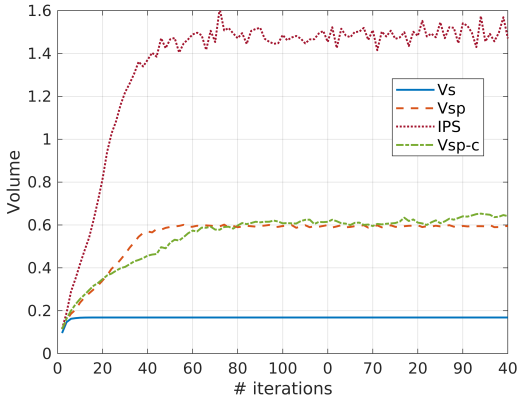
The goal of applying the algorithms to a system which does not have a ROA centered at the origin is meaningful, because it reflects the condition adopted by each algorithm to enlarge the ROA subset. Figure 7.3(a) shows that the IPS algorithm obtains a better subset in comparison with the others, as expected because the enlargement condition used relies on the constraint of the ROA subset by the previous subset.

It is relevant to point out that the IPS stopped due to feasibility issues at iteration number 18. Additionally, the addition of a second shape function reveals to give a better estimate as seen in Figure 7.3(b), by checking the subset volume obtained by the Vsp-c algorithm in comparison with the Vsp algorithm; the ratio between volumes of these two algorithms is close to two.

Figure 7.4(a) and 7.4(b) present results from applying the previous stated algorithms to system (S4) which has a bounded ROA that is non-convex.



(a) Estimated ROA subset.

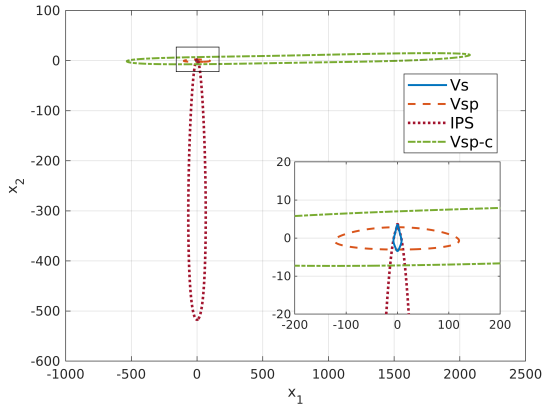


(b) Volume evolution over iterations of the ROA subset.

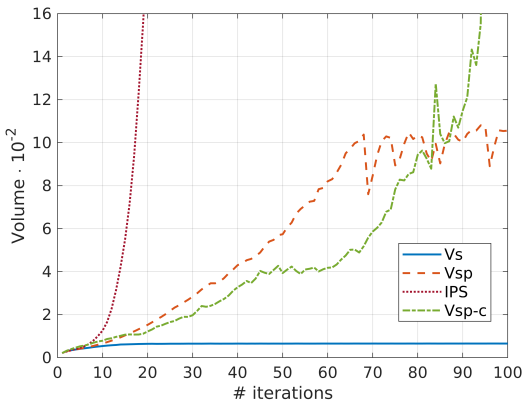
Figure 7.4: Results of algorithm testing for system (S4).

As expected, the Vsp and Vsp-c algorithms have similar results, the additional shape function present in the Vsp-c algorithm does not improve the estimate of the ROA due to the non-convexity and symmetry of the true ROA. In this system, it is observed the presence of oscillations due to numerical errors and some inherent errors in the volume calculation which was performed by a Monte Carlo simulation.

The following results concern system (S5) which is a globally asymptotically stable system with the equilibrium point at the origin, but it does not admit a polynomial LF of any degree [71] and for this reason it is interesting to test the presented algorithms because they only consider polynomial dynamics and local polynomial LFs.



(a) Estimated ROA subset.



(b) Volume evolution over iterations of the ROA subset.

Figure 7.5: Results of algorithm testing for system (S5).

Figures 7.5(a) and 7.5(b) show that the Vs algorithm has a poor performance achieving low values for the ROA subset volume in comparison with the others. This is due to the non-adaptability of this algorithm to the initial conditions. The IPS algorithm has the best performance with respect to the estimated volume and the rate in which it converges, in other words it has the highest rate of convergence of its subset. Both Vsp and Vsp-c algorithms suffer from oscillations in their estimate volume over iterations, and it is relevant to remark that the Vsp-c algorithm has at the end of the 100 iterations a

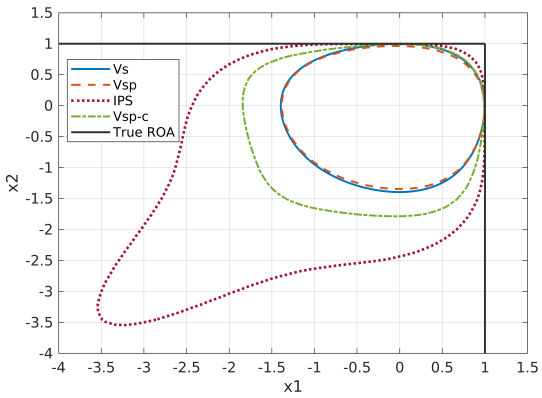
higher volume than the Vsp algorithm but until around the 80th iteration it had a worse estimate. This is likely due to the way the shape functions and LF evolve over iterations. It is relevant to point out the way in which the estimates of the ROA evolved. We observe that for the Vsp and Vsp-c the estimate enlarges predominately in the x-axis. By contrast the estimate obtained by the IPS algorithm enlarges predominately in the y-axis.

The final planar dynamical system we study here corresponds to system (S6) which has an unbounded ROA characterized by the semi-algebraic set

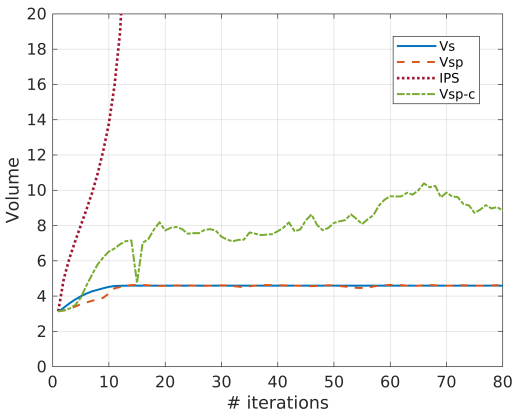
$$\Omega = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1 \leq 1,\ x_2 \leq 1\}$$

with $\mathbf{x} = [x_1 \quad x_2]^\top$.

It is important to point out that not only system (S6) has an unbounded ROA but also that its boundary is not continuously differentiable. At point $\mathbf{x} = [1 \quad 1]^\top$, it does not exist a derivative of the curve representing the boundary of the ROA. The results from applying the algorithms to system (S6) are presented in Figure 7.6(a) and 7.6(b).



(a) Estimated ROA subset.



(b) Volume evolution over iterations of the ROA subset.

Figure 7.6: Results of algorithm testing for system (S6).

The ROA of system (S6) is not centered at the origin, and it shows how it affects the performance of the Vs and the Vsp algorithms, that is, those two algorithms have similar results and display a poor performance due to constraining the subsets with an ellipsoid centered at the origin. Additionally, the IPS algorithm once again has the best result and the Vsp-c algorithm reaches a better estimate than the Vsp algorithm due to enlarging the subset with an additional shape function not centered at the origin. In other words, it corresponds to constraining the subset by an ellipsoid not necessarily centered at the origin but at the centroid of the estimated ROA.

It is relevant to point out that the behavior over iterations of the Vsp-c is quite oscillatory due to the rough calculation of the centroid which affects the update of the shape function, and also it is due to possible numerical problems with the LMI solver.

7.2 Computational statistics

From experience and application of the iterative bilinear sum-of-squares algorithms presented in Chapter 5, we observe that each algorithm is prone to computational problems such as numerical and feasibility issues of the solver. This leads either to some odd behaviors, such as decreasing the volume when it is imposed only an increase or to an early stop. For this reason, this section is intended to study such issues and give an idea on how each algorithm performs on arbitrary systems.

Inspired by [27], we selected a system where its ROA has an analytical expression and can be directly extracted from the system dynamics. Consider

$$\dot{\mathbf{x}} = -\mathbf{I}\mathbf{x} + (\mathbf{x}^\top \mathbf{B}\mathbf{x})\mathbf{x} \tag{7.1}$$

with $\mathbf{x} \in \mathbb{R}^n$ and a symmetric matrix $\mathbf{B} \succ 0$ of size $n \times n$.

The ROA of system (7.1) is an ellipsoid characterized by the set $L^\circ(\mathbf{x}^\top \mathbf{B}\mathbf{x}, 1)$ and it has a LF that has its time derivative negative over the ROA described by $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{B}\mathbf{x}$. In Appendix A, we present a proof for the previous statements.

So to study how well each algorithm behaves in average, we use system (7.1) with different dimensions ranging from 2 to 6, and for 100 randomly generated positive definite matrices $\mathbf{B}$ with appropriate size. Some relevant considerations were taken while choosing matrix $\mathbf{B}$, we observed that each algorithm behaves differently for different eigenvalues of matrix $\mathbf{B}$. More specifically, we noticed that each algorithm depends on the matrix maximum eigenvalue ratio.

Definition 7.1 (Maximum eigenvalue ratio)**.** *For a matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. Let $\lambda_{max} = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$ and $\lambda_{min} = \min\{|\lambda_1|, \ldots, |\lambda_n|\}$. The maximum eigenvalue ratio is given by $\lambda_{max}/\lambda_{min}$.*

We observed that for high values it is more likely to encounter computational issues, leading to an early stop of the algorithm. Thus, we chose $\mathbf{B}$ with a maximum eigenvalue ratio ranging from 1 to 900, because within this range the frequency of encountering computational issues is almost constant with respect to any admissible matrix $\mathbf{B}$.

Due to numerical factors that the solvers are sensible to, we rescaled each system to have always a ROA of unitary volume. The results of the average volume achieved by each algorithm is presented in Table 7.1.

Table 7.1: Normalized average volume of the subset of the ROA obtained by applying each algorithm to system (7.1) considering different system sizes. Average results from testing on 100 randomly generated systems.

	Average Volume				
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
Vs	0.156	0.098	0.072	0.056	0.052
Vsp	0.998	0.936	0.997	0.966	0.998
IPS	0.976	0.916	0.843	0.702	0.739
Vsp-c	0.988	0.989	0.987	0.997	0.997

We observed from Table 7.1 that the Vs gives extremely poor results as expected, because in contrast with the other algorithms it has no adaptive shape function. In other words, the initial guess has a big influence on the estimate of the ROA obtained.

The other three algorithms, Vsp, IPS and Vsp-c do not have the same issue as the Vs algorithm since they are adaptive to the initial guesses, that is, the parameters chosen at the beginning will change over iterations and a less dependence on the initial guesses occurs.

From Table 7.1, we observe that the Vsp and Vsp-c algorithms converge close to the true ROA for any of the dimensions considered and the volume achieved is always above 0.9. This contrasts with the IPS algorithm, that exhibits a decrease in performance for higher dimensions and this is mostly due to numerical problems that led to an early stop.

For the same set of runs, we also retrieved the frequency of each algorithm to end in a normal stop. We consider a normal stop, when no numerical problems occur and the termination condition is achieved. The termination condition here chosen was when the volume increase from one iteration to the other is less than 0.01 times the previous volume calculated, i.e., the termination condition indicates convergence of the algorithm. Table 7.2 presents the frequency of normal stop for each algorithm for the same set of runs previously presented.

Table 7.2: Ratio of number of runs that stopped under the condition imposed and the number of total runs performed considering systems defined in different dimensions.

	Frequency of normal stop				
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
Vs	1	1	1	1	1
Vsp	1	0.83	1	0.8	1
IPS	0.96	0.43	0.3	0.2	0.05
Vsp-c	0.93	0.96	1	1	0.9

Table 7.2 backs the previous statement in relation to the IPS algorithm, that is, the IPS algorithm performs poorly for higher dimensions due to numerical issues that led to an early stop, we can see that for $n = 2$ the frequency of a normal stop is of 0.96 and for $n = 6$ the value is of 0.05, having as a consequence the volume shown in Table 7.1, obtaining in average 0.976 and 0.739, respectively. All the other algorithms, Vs, Vsp and Vsp-c algorithms do not fall frequently into numerical or feasibility problems for the test performed in comparison to the IPS algorithm.

Even though system (7.1) is a good system to evaluate and compare each algorithm, it is quite simple and only represents system of polynomial degree three, having always a ROA represented by a quadratic polynomial. We propose a generalization of the system, allowing to have regions of attraction other than just ellipsoids and also to study systems of different polynomial degrees, other than just systems of polynomial degree three.

Consider the system

$$\dot{\mathbf{x}} = -\mathbf{I}\mathbf{x} + (\mathbf{z}^\top \mathbf{B}\mathbf{z})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n \tag{7.2}$$

with $\mathbf{B} \succ 0$, $\mathbf{z} = [z_1, \ldots, z_m]^\top$ and z_i is a monomial of $\mathbf{x}$ of degree bigger than 0, that is, $\delta^+(z_i) \geq 1$ for $i = 1, \ldots, m$. It is clear that the set $L(\mathbf{z}^\top \mathbf{B}\mathbf{z}, 1)$ is an invariant set of system (7.2) because it is a bounded set and any point $\mathbf{x}$ satisfying $\mathbf{z}^\top \mathbf{B}\mathbf{z} = 1$ is a stationary point.

However, the question is if the set $G = L^\circ(\mathbf{z}^\top \mathbf{B}\mathbf{z}, 1)$ is also the exact ROA of system (7.2). If we consider $V(\mathbf{x}) = \mathbf{z}^\top \mathbf{B}\mathbf{z}$ as a LF, we verify that the set G is the ROA of system (7.2) if

$$\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}} \mathbf{z})\mathbf{x} > 0 \tag{7.3}$$

holds for any point inside $G \setminus \{0\}$. The corresponding proof and some specific case scenarios are presented in Appendix A.

Condition (7.3) gives a condition to guarantee that G is the ROA of system (7.2). We observed through testing on multiple configurations that if the condition is not verified it does not mean that G is not the ROA.

Remark 1. *System (7.2) will always give a closed ROA and the degree of the polynomial dynamical system is always odd for any admissible matrix $\mathbf{B} \succ 0$ and vector of monomials $\mathbf{z}$ of degree greater than 0. Different ROA for various configurations of the vector $\mathbf{z}$ are presented in Figure 7.7.*

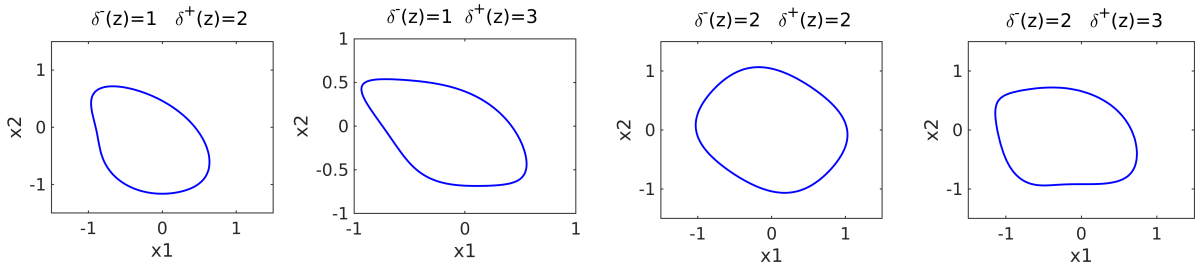


Figure 7.7: Different ROA taking into account various system dynamics configurations of system (7.2).

To study for different configurations, we consider planar systems, $\mathbf{x} \in \mathbb{R}^2$, with different vectors $\mathbf{z}$, that is, we set different values for the minimum and maximum degree of each monomial in vector $\mathbf{z}$, $\delta^-(z)$ and $\delta^+(z)$, respectively.

Table 7.3 presents results for the average volume reached for each algorithm for different system configurations. Fifty different runs were executed for each configuration and the volume of the exact ROA is defined to be unitary. It is relevant to state that in this case no consideration over the eigenvalues of matrix $\mathbf{B}$ were taken.

Table 7.3: Normalized average volume of the ROA subset achieved applying each algorithm to different system configurations.

$\delta^-(\mathbf{z})$	$\delta^+(\mathbf{z})$	**Vs**	**Vsp**	**IPS**	**Vsp-c**
1	1	0.2809	0.7747	0.8125	0.8703
1	2	0.3725	0.9347	0.9776	0.9356
1	3	0.3725	0.5117	0.9815	0.4641
2	2	0.7708	0.9271	0.8931	0.9310
2	3	0.4701	0.5397	0.7145	0.6110
3	3	0.7461	0.8220	0.8028	0.8183

The results from Table 7.3 are quite complicated and stating the existence of a better overall algorithm is impossible. Still, some observations can be done, we can observe that the Vs performs the worse in comparison to any other algorithm for any system configuration. As expected of the IPS, it presents in general better results, this is because it differs from the other algorithms by finding a positively invariant set and not a sublevel set of the LF that represents an estimate of the ROA.

We can say that the Vsp-c algorithm performs better than the Vsp algorithm in general, except for a single configuration. This is due to the fact that the Vsp-c algorithm constrains the estimate more and consequently allows for better results. Also, the Vsp-c algorithm is not entirely dependent on the quadratic terms of the LF estimated each iteration, unlike Vsp algorithm that each iteration gets a new shape function from the quadratic terms of the LF estimated; there can be a situation where the LF estimated does not have any quadratic terms, leading the Vsp algorithm to stop.

Additionally, we observed that some normal stops did not result in a good estimate, we suppose that it is due to a local optima that the algorithm fell into or that the estimate is being constrained aggressively in some regions, and consequently it leads to an extremely slow increase in the estimate's volume.

7.3 Time computational complexity

The iterative bilinear sum-of-squares algorithms presented in Chapter 5 solve the optimization problem of finding the largest subset of the ROA through a sequence of optimization steps. The optimization problem is divided into a set of sub-problems that if solved iteratively converges to the solution of the original optimization problem.

Each of the optimization steps are a type of optimization task known as generalized SOS program, already presented in Chapter 4. It is well known that SOS programming has scalability issues, for instance, if we consider $p(\mathbf{x}) \in \Sigma[\mathbf{x}]$ of degree $2k$ with $\mathbf{x} \in \mathbb{R}^n$, the size of its Gram matrix $\mathbf{Q}$ is $\binom{n+k}{k} \times \binom{n+k}{k}$. Thus, an increase in the dimension n leads to a big increase in the size of the Gram matrix and consequently the number of decision variables.

Therefore, the time it takes to complete a single iteration, that is, one sequence of steps, depends on the system size n but also on the constraints and decision variables associated with each step. In particular, the computational complexity of the solver SeDuMi [66] is solely dependent on the size of the given LMI [59] that is created from the SOS constraints by using Sosopt [67], which is a function of the aforementioned parameters and the number of decision variables.

To be more specific, the asymptotic computational complexity of SeDuMi including main and inner iterations is in $\mathcal{O}(p^2q^{2.5} + q^{3.5})$, with p as the number of decision variables and q as the number of rows of the LMI, while for example the asymptotic computational complexity of the LMI solver used in the LMI Control Toolbox of Matlab [72] is in $\mathcal{O}(p^3q)$. In both solvers, the dependence is only on the number of decision variables and the LMI size and so it is interesting to understand how each iterative bilinear SOS algorithm that estimates the ROA of a polynomial dynamical system depends on these parameters.

We can have a clear idea of the number of decision variables per step, because it is a predefined parameter in that we choose the degree of the auxiliary polynomial decision variables, such as the LF or the s-multipliers to run the algorithm. In contrast, it is extremely hard to have a clear idea of the number of rows of the generated LMI in each step, because it's highly dependent on the polynomial dynamical system configuration, not only it's minimum and maximum degree but also the monomials that it has and other auxiliary functions.

In such manner we intend to study for a general polynomial dynamical system its computational time complexity, through some observations and specific case scenarios. Mostly we are interested in checking the requirements in the number of decision variables in each step as this parameter is the only one that can be predicted for a general dynamical system of an arbitrary degree.

Taking into consideration the above statement, we assembled for each algorithm, the number of decision variables in each step that they have depending on the degrees of the polynomial decision variables present.

Let $s \in \mathbb{R}[\mathbf{x}]$ be a polynomial decision variable with respect to $\mathbf{x} \in \mathbb{R}^n$ and of degree d, that is $\delta^+(s) = d$ and $\delta^-(s) = d$, the number of decision variables or coefficients is given by

$$p(n,d) = \binom{n+d-1}{d} \tag{7.4}$$

With Expression (7.4), we can write the total number of decision variables for each step taken for all the iterative bilinear SOS algorithms presented in Chapter 5, by checking the degrees of the decision variable functions.

First, for the Vs and Vsp algorithms, which have the same iterative structure, we have the following expressions for the number of decision variables for each of the steps:

Number of decision variables for Vs and Vsp algorithms

$\sum_{d=\delta^-(s_0)}^{\delta^+(s_0)} p(n,d)$	$\sum_{d=\delta^-(s_1)}^{\delta^+(s_1)} p(n,d)$	$\sum_{d=\delta^-(V)}^{\delta^+(V)} p(n,d)$
1º Bisection Step	**2º Bisection Step**	**Convex step**

The expressions for the number of decision variables for the Vsp-c algorithm are equal to the ones for the Vs and Vsp algorithm, but it has an additional bisection step.

Number of decision variables for Vsp-c algorithm

$\sum_{d=\delta^-(s_0)}^{\delta^+(s_0)} p(n,d)$	$\sum_{d=\delta^-(s_1)}^{\delta^+(s_1)} p(n,d)$	$\sum_{d=\delta^-(s_2)}^{\delta^+(s_2)} p(n,d)$	$\sum_{d=\delta^-(V)}^{\delta^+(V)} p(n,d)$
1º Bisection Step	**2º Bisection Step**	**3º Bisection Step**	**Convex Step**

Unlike the aforementioned algorithm, the IPS algorithm does not have a convex step, instead it has two bisections steps with a higher number of polynomial decision variables in each compared to the bisection of the other algorithms.

Number of decision variables for IPS algorithm

$\sum_{i=0}^{2} \sum_{d=\delta^-(s_i)}^{\delta^+(s_i)} p(n,d) + \sum_{d=\delta^-(V)}^{\delta^+(V)} p(n,d)$	$\sum_{d=\delta^-(R)}^{\delta^+(R)} p(n,d) + \sum_{d=\delta^-(V)}^{\delta^+(V)} p(n,d) + \sum_{d=\delta^-(s_3)}^{\delta^+(s_3)} p(n,d)$
1º Bisection Step	**1º Bisection Step**

In the literature the above expressions are not shown or even worked upon because typically each author conceives an algorithm and applies to a specific minor set of problems and does not take into account the diversity of set of systems that the algorithm can be used in. Also, these expressions are quite hard or even sometimes misleading because they depend on the user choice for the s-multipliers degree and also for other functions degree, for example function $V(\mathbf{x})$ and $R(\mathbf{x})$ on the IPS algorithm.

Just as an example to check the number of decision variables, we consider systems of sizes ranging from 2 until 8 of maximum polynomial degree of four. To choose the corresponding s-multipliers $s_i(\mathbf{x})$, we resorted to the procedure presented in Chapter 6, which as stated before gives somewhat of a minimum and necessary size for each s-multipliers so that the algorithm can run without encountering feasibility issues and gives also good results. In Table 7.4, the corresponding results are presented.

Table 7.4: Number of decision variables when applying the algorithms to a polynomial dynamical system of maximum degree four and considering the estimate of the ROA characterized by a semi-algebraic set with a single polynomial inequality of degree four.

	Vs & Vsp			**IPS**		**Vsp-c**			
n	b1	b2	c	b1	b2	b1	b2	b3	c
2	15	6	12	18	9	15	6	6	12
3	35	10	31	61	38	35	10	10	31
4	70	15	65	160	112	70	15	15	65
5	126	21	120	360	271	126	21	21	120
6	210	28	203	729	577	210	28	28	203
7	330	36	322	1365	1121	330	36	36	322
8	495	45	486	2404	2031	495	45	45	486

It is relevant to point out that convex steps are less laborious than bisections steps with the same number of decision variables and LMI, because a bisection is a sequential computation of convex steps.

From Table 7.4, it is evident that the IPS algorithm is less suited for large systems than the others because it has a larger increase in number of decision variables for systems in higher dimensions, and consequently this corresponds to an increase in computation time of one iteration. The Vsp-c has an additional bisection step, and consequently it will take more time to perform one iteration compared to the Vs and Vsp.

The aforementioned statements and observations can be misleading, because they only take into consideration global aspects of the algorithms, such as the number of decision variables. If we consider the total time spent by each algorithm to converge to a solution, it is revealed that the total time is not only dependent on the number of decision variables and on the LMI size, we can observe the results from Chapter 7.1, where different behaviors were shown, for example the IPS typically has a high rate of convergence, evolving its estimate volume faster to the end value compared to the other algorithms. That is to say, that even though the IPS is the algorithm which needs more resources to perform a single iteration, it has the advantage of evolving the estimate volume faster per iteration and so sometimes it can converge faster to its end value compared to the rest.

7.4 Effect of initial conditions

A relevant aspect of the iterative bilinear sum-of-squares ROA estimation algorithms is the necessity to select initial guesses, for example it may be required an initial guess of a LF or some guesses related to auxiliary functions. It is shown in the literature, for instance in [50] that for the Vs algorithm the initial value assigned to the parameters, such as the shape function, has an influence in the obtained ROA subset.

For the Vs algorithm, the LF selected at the beginning needs to give a feasible solution at least in the first iteration, so authors typically used a quadratic LF obtained by linearizing the system around the equilibrium point. The method used to get an initial guess of the LF is based on the linearized model of the system because we can get with ease a quadratic LF from it. This method is described in Algorithm 5

Algorithm 5. *Method to obtain a quadratic LF from the linearized system:*

1. *Select a symmetric positive definite matrix* $\mathbf{Q}$*.*

2. *Get the linearized system from system 2.1, resulting in* $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ *with* $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big|_{\mathbf{x}=0}$*.*

3. *Solve* $\mathbf{A}^\top \mathbf{B} + \mathbf{B}\mathbf{A} + \mathbf{Q} = 0$*, with* $\mathbf{B}$ *as a matrix decision variable.*

4. *A possible quadratic LF of system 2.1 is given as* $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{B}\mathbf{x}$*.*

It is important to notice that this method to obtain an initial guess might not work when the linearized system is not asymptotically stable, in other words, the linearized system around the equilibrium point considered does not have all eigenvalues with strictly negative real part which results in a failure of Algorithm 5.

In some literary works, the initial guess of the shape function is independent on the initial choice of the LF. Typically, a random choice of the shape function is used such as $\mathbf{x}^\top \mathbf{x}$. We established the initial value of the shape function equal to that of the quadratic LF's initial guess because the shape function should reflect the major directions of the system dynamics to allow for a better ROA estimate.

To illustrate that the initial guesses have an influence on the ROA subsets obtained, Figure 7.8 shows four subsets of the ROA obtained by applying Vs algorithm to system (S1) with guesses for the LF using Algorithm 5 with different choices of $\mathbf{Q}$. The produced quadratic LFs are

$$V_1(\mathbf{x}) = 0.04123x_1^2 + 0.02157x_1x_2 + 0.07346x_2^2 \qquad V_2(\mathbf{x}) = 0.24077x_1^2 + 0.40740x_1x_2 + 0.38764x_2^2$$
$$V_3(\mathbf{x}) = 0.11723x_1^2 + 0.11464x_1x_2 + 0.07764x_2^2 \qquad V_4(\mathbf{x}) = 0.35348x_1^2 + 0.28565x_1x_2 + 0.39056x_2^2$$

with $\mathbf{x} = [x_1 \quad x_2]^\top$.

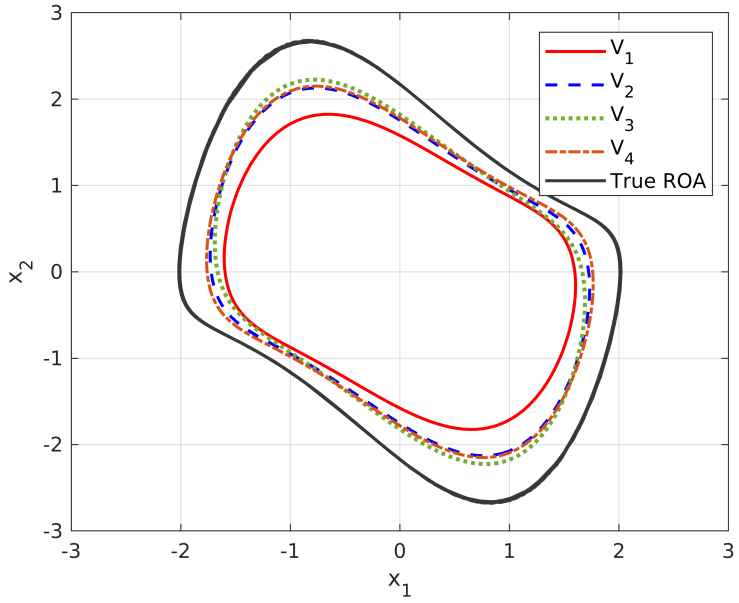


Figure 7.8: Example of the effect of initial guesses for the Vs algorithm.

Figure 7.8 clearly shows that the choice of the initial LF and shape function has an influence on the obtained ROA subset. Apart from this result in relation to the Vs algorithm, in the literature no clear study is presented for the other algorithms, that is, no study exists to check the robustness and sensitivity to initial guesses of the presented iterative bilinear ROA estimation algorithms.

To study how each algorithm performs in average for different initial guesses, a statistical approach was performed by using system (S1), that is characterized by having a closed ROA. For this system, the method to obtain a initial guess of the LF can be performed by using Algorithm 5, because its linear model is Hurwitz.

In the proposed statistical study, we used the procedure to select the s-multipliers presented in Chapter 6, and we considered a termination rule which consisted in evaluating the variation in volume of the obtained subsets in consecutive iterations. The algorithm comes to a halt if the variation in volume from consecutive iterations is less than 1% or if the number of iterations reached 100.

Additionally, in this test we verify what is the influence of the initial conditions for different degrees of the sublevel set that describes the ROA estimate. In general a higher degree implies a better estimate, so this test will check if the initial conditions can have a big effect on this general observation.

Figure 7.9 presents the average, minimum and maximum values of the volume achieved for each algorithm when applied to system (S1), considering different degrees for the polynomial sublevel set that describes the estimate of the ROA.

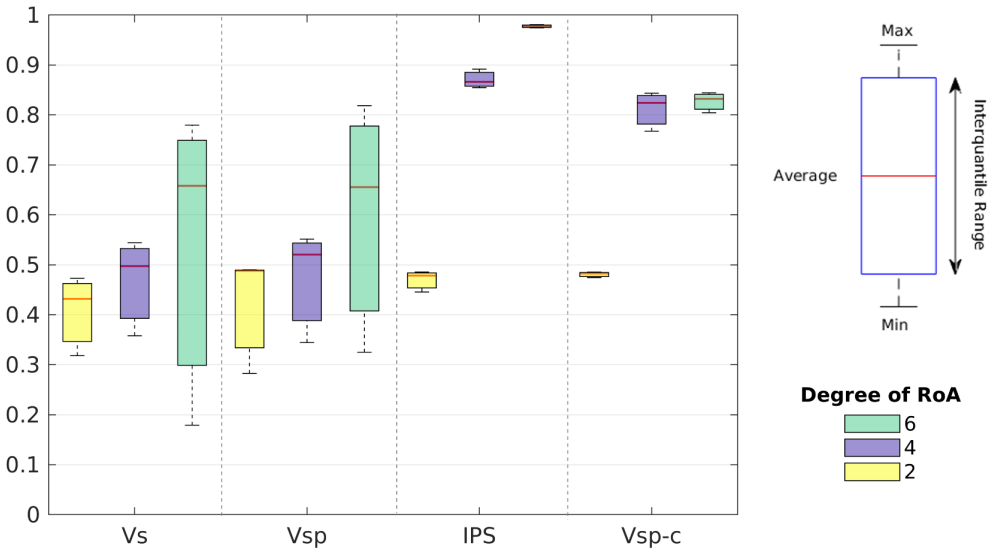


Figure 7.9: Statistics on the volume of the converged ROA estimate for various polynomials degrees describing the ROA (2, 4, 6). The results arise from 100 randomly generated runs. The horizontal red line (——) represents the average volume for each algorithm with a specific degree of the polynomial that characterizes the subset of the ROA.

Figure 7.9 shows that the IPS algorithm is extremely resilient to initial setup when applied to system (S1) for any degree selected for the ROA estimate. This observation was foreseen, due to the enlargement condition present in this algorithm, which forces the following estimate to contain the whole previous estimate.

In contrast, the Vs algorithm is the most sensible to initial guesses for any polynomial degree chosen for the ROA. This occurs due to not having any adaptive criteria for its initial guess of the shape function, making the algorithm highly dependent on the initial guess.

In comparison to the Vs and Vsp algorithms, the Vsp-c algorithm shows a better average result and less sensitivity to initial guesses, as a result of the additional enlargement constraint added by means of a second shape function.

Additionally, we see that increasing the degree of the polynomial sublevel set that describes the ROA does not guarantee a better estimate, as seen for the Vs and Vsp with a ROA of degree 6, we check a wider range of ROA volume for this degree. This occurs mostly due to the enlargement condition used in these algorithms, which is not strict enough to guarantee an increase in the estimate, because they are constraining the subset of the ROA by ellipsoids and not the entire previous estimate and also due to the occurrence to feasibility issues.

7.5 Systems in high dimensions

To verify the behavior of each algorithm to systems other than planar systems, some relevant systems in higher dimensions are considered and worked upon. The following three dynamical systems and respective results are presented and some characteristics and limitations of each iterative bilinear sum-of-squares ROA estimation algorithm are shown.

7.5.1 Example I – Simple system in $\mathbb{R}^3$

The first system to illustrate the performance of each algorithm to systems in higher dimensions is a dynamical system defined in $\mathbb{R}^3$ with polynomial dynamics of maximum degree two. This system has an unbounded ROA and to the current knowledge it has no analytical description of its ROA.

$$\dot{\mathbf{x}} = \begin{bmatrix} -x_1 + x_2 x_3 \\ -x_2 + x_1 x_2 \\ -x_3 \end{bmatrix} \tag{S7}$$

To give an insight into the true ROA of this system, some trajectories using RTM was performed, and the results are shown in Figure 7.10, where a three-dimensional plot of the trajectories belonging to the ROA and two additional plots with projection into the $x_1 - x_2$ plane and the $x_1 - x_3$ plane is presented.

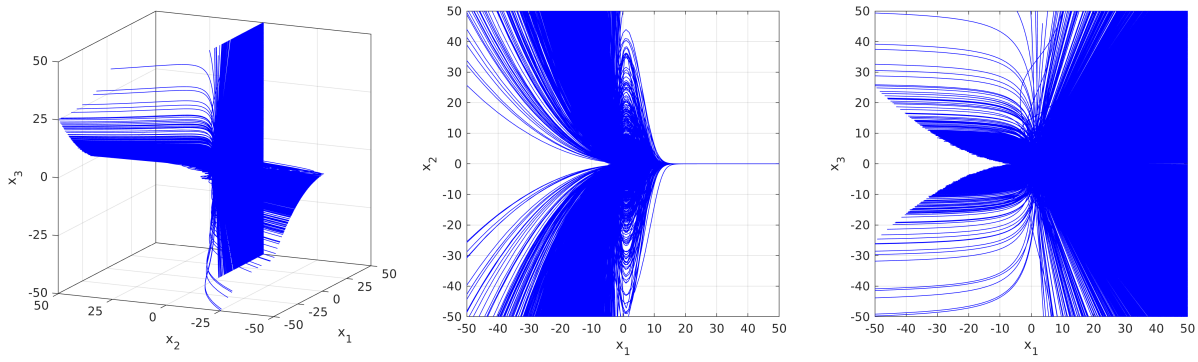


Figure 7.10: Trajectories generated by applying RTM to system (S7).

Figure 7.10 shows that the ROA shape of system (S7) is complicated and irregular; relying on trajectory generation gives difficult results to interpret and explore, so relying on the algorithms presented can give a clearer idea of the ROA.

The comparative results from applying algorithms Vsp, Vsp-c, Vs and IPS are presented in Figure 7.11, where each estimate of the ROA is characterized by a single polynomial inequality of degree four. As a remark, the y-axis is in logarithm scale due to disparity of the results obtained.

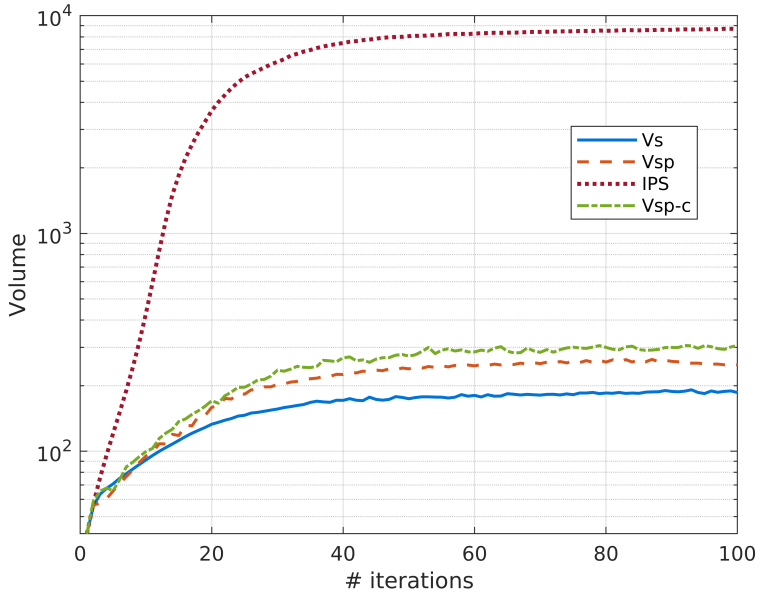


Figure 7.11: Evolution of the ROA estimate volume for system (S7) over iterations.

From the results present in Figure 7.11, the Vs algorithm has the worst performance because the volume of the semi-algebraic set obtained at the end is less than any of the other algorithms. The IPS algorithm has the best performance, achieving values of volume of the estimate of the ROA more than 10 times the volume reached by the other algorithms. Additionally, it is observed that the Vsp-c algorithm has a slight better performance in terms of volume in relation to the Vsp algorithm.

7.5.2 Example II – Lorenz system

The Lorenz system is one of the most studied dynamical systems in the literature [53, 73–75]. This system has a chaotic behavior, that means that a small change in the initial condition can alter the outcome significantly. The specific Lorenz system chosen is presented in equation (S8), and it was modified such that a local asymptotic equilibrium point is located at the origin.

$$\dot{\mathbf{x}} = \begin{bmatrix} -10x_1 + 10x_2 \\ -x_1x_3 + x_1 - x_2 - 3x_3 \\ x_1x_2 + 3x_1 + 3x_2 - x_3 \end{bmatrix} \tag{S8}$$

In Figure 7.12, some trajectories generated by the RTM of the Lorenz system are presented, revealing the chaotic behavior stated before and that the ROA is unbounded and has a strange format. Additionally, it is relevant to remark that there is no analytic expression for the ROA associated with the equilibrium point.

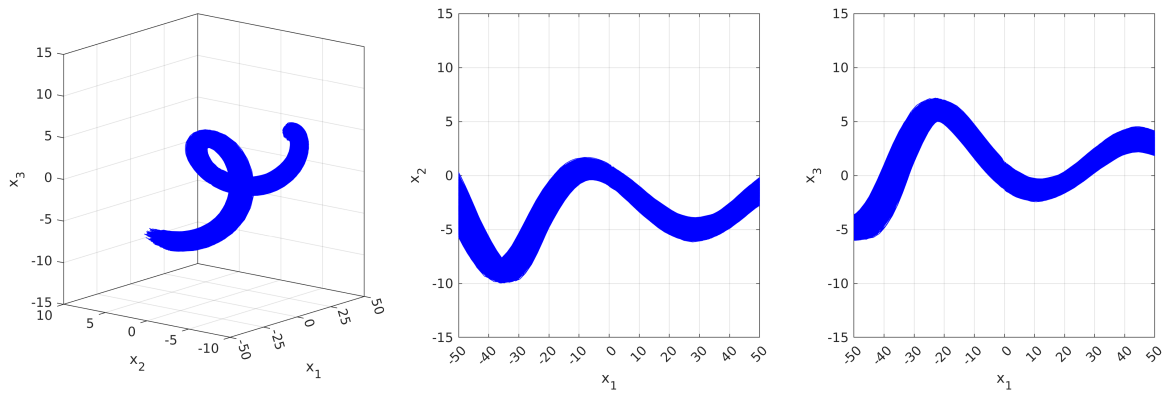


Figure 7.12: Trajectories generated by applying RTM to system (S8).

The results from applying each algorithm presented in Chapter 5 to system (S8) are presented in Figure 7.13.

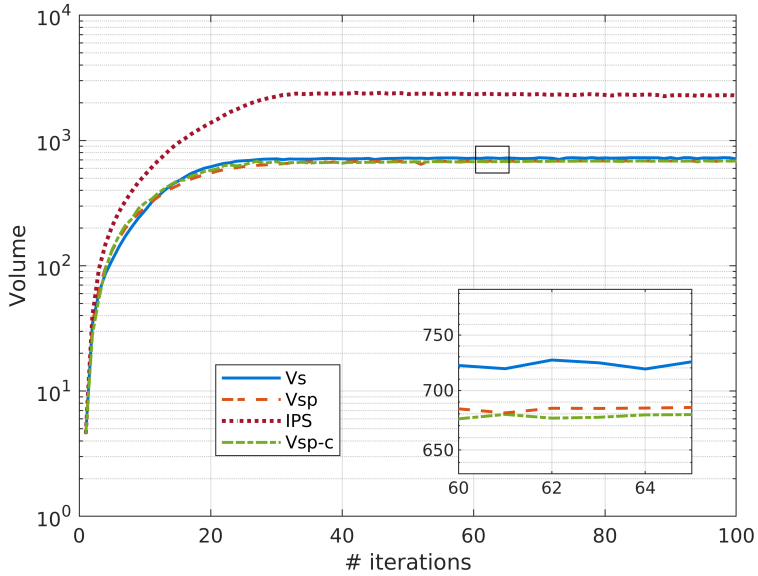


Figure 7.13: For the Lorenz system (S8), ROA estimate volume evolution over iterations.

From the results shown in Figure 7.13, the IPS achieves a better estimate of the ROA and unlike some examples presented before for planar systems, the Vs reaches a slightly better estimate than the Vsp and Vsp-c. In this specific case, the Vsp and the Vsp-c have the same results for the ROA subset obtained.

7.5.3 Example III – Inverted 2-link pendulum

The single pendulum or simply inverted pendulum is a well known and intensively worked dynamical system, as it is a frequent physical model used to verify and test some control theory and techniques. An extension of the single pendulum and that reveals to be a very popular dynamical system among the scientific community is the inverted N-link pendulum, since it is an underactuated system with nonlinear, unstable and minimum phase behaviors [76].

Unlike the single pendulum, for $N > 1$ the systems exhibits an increased difficulty to control in that it is underactuated and with minimum phase. Additionally, it is shown that the ROA becomes arbitrarily small as the number of links increases [77] and that the system is very sensitive to disturbances, joint friction and measurement noise. For the case with a single input and $N > 1$, the linearized system around the equilibrium point is not controllable, leading usual linear control methods without a guarantee of stability of the system. Nevertheless, it is still possible to use linear control laws, but it is of extreme concern to know the limits of the control law, in other words, the resulted ROA [78].

For the reasons aforementioned, it is quite interesting and relevant to observe the performance of each ROA estimation algorithm to this type of system. In the presented work, a polynomial approximation of a double inverted pendulum dynamics with a single input and with a linear controller is considered as a case study to verify the outcome of applying the algorithms to a highly nonlinear system in $\mathbb{R}^4$. The approximate polynomial system is described in Appendix B for a 2-link inverted pendulum on a moving car and denoted (B.3).

The ROA of this system is unbounded, as observed by applying a RTM to the controlled system, resulting in the trajectories presented in Figure 7.14.

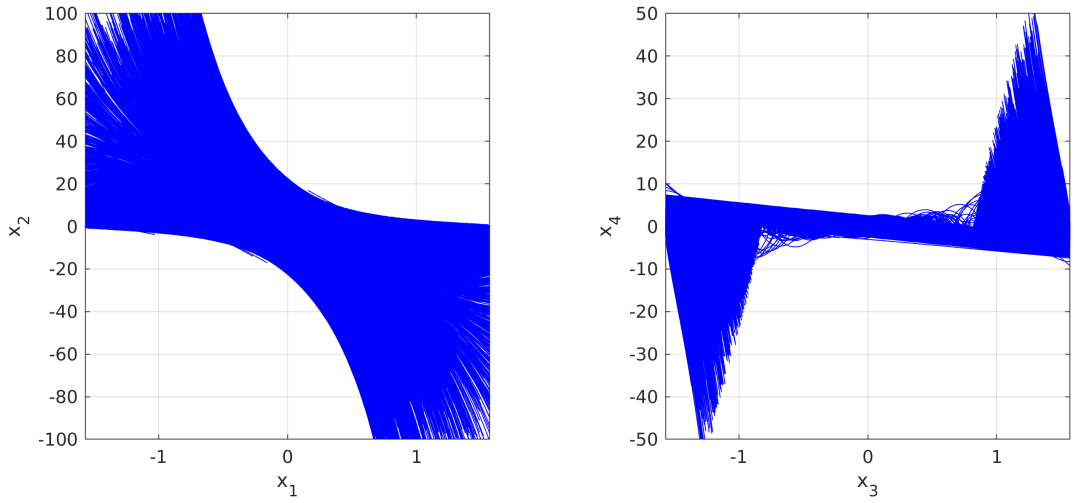


Figure 7.14: Trajectories generated by the RTM for the approximated polynomial pendulum system.

The results of the evolution of ROA subsets volume over iterations estimated by each algorithm presented in Chapter 5 are shown Figure 7.15, considering a subset of the ROA characterized by a polynomial of degree four.

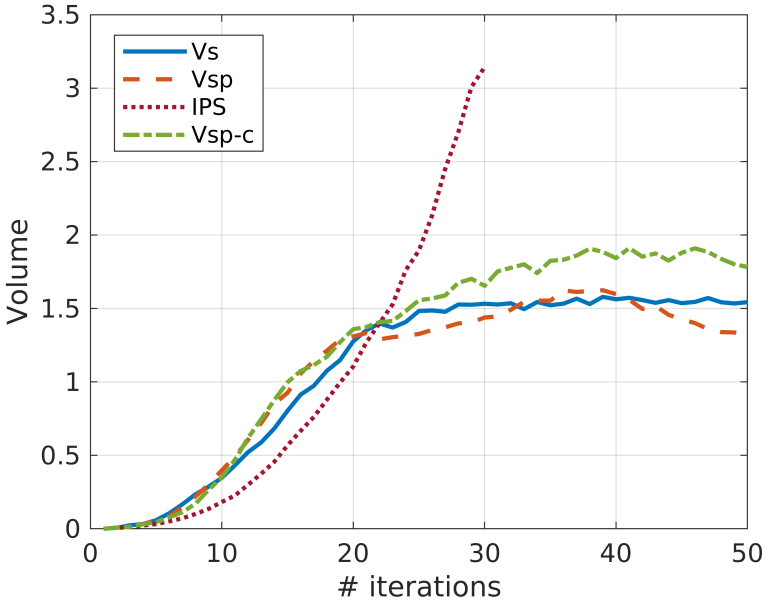


Figure 7.15: Volume of the ROA estimate over iterations

Figure 7.15 shows that although it takes some time for the IPS algorithm to initially raise its estimate volume, it reaches a better estimate over a long number of iterations, this is due to the higher number of decision variables in each step but also due to the enlargement condition imposed by it. In other words, the IPS algorithm despite having a better result asymptotically, it exhibits a worse ROA estimate than the others for this specific case study in $\mathbb{R}^4$ for a short number of iterations.

The remainder algorithms perform similarly, achieving values for the volume of the ROA subset close to each other, remarking that the Vsp-c achieved a slightly better result.

Table 7.5 presents the average time per iteration for each algorithm when applied to the 2-link inverted pendulum, considering a subset of the ROA of degree four.

Table 7.5: Average computation time in each iteration for the pendulum system by each algorithm.

	Vs	**Vsp**	**IPS**	**Vsp-c**
Average time per iteration [s]	13.708	15.088	41.861	15.995

From the results in the average time per iteration taken by each algorithm shown in Table 7.5, the Vs, Vsp and Vsp-c algorithms take similar amount time to perform a single iteration and by contrast the IPS algorithm takes almost 2.5 times more to perform a single iteration in average. Even though an algorithm spends more time to compute in average each iteration, volume of the estimate can have a higher rate of increase, and it may converge faster, although each iteration takes more time.

CHAPTER 8

Novel Approach

This chapter presents a novel approach to estimate the ROA which does not rely on Lyapunov's theory, instead an approach is developed using the Bendixson-Dulac criterion for planar systems and extensions of it to higher dimensional systems. Three different iterative bilinear sum-of-squares algorithms are presented and discussed, one which only works for planar dynamics and two others that work for any dimension, even though some restrictions are imposed to them to enable the formulation as an iterative bilinear sum-of-squares algorithm.

8.1 Bendixson-Dulac criterion

A feature of some dynamical systems is the presence of periodic orbits, in particular, for planar dynamics the problem of checking the existence and number of this type of trajectories is related to one of the famous Hilbert's problems [79], which is until now still unsolved. For planar systems a well known criterion named Bendixson-Dulac criterion gives a condition for the non-existence of periodic orbits on an autonomous system over a simply connected set. It is stated as follows.

Theorem 8.1 (Bendixson-Dulac (BD) criterion [80])**.** *Consider system (2.1) with* $\mathbf{x} \in \mathbb{R}^2$*. Let* $\phi(\mathbf{x})$ *be a positive scalar continuously differentiable function on a simply connected region* $D \subseteq \mathbb{R}^2$*. If*

$$\operatorname{div}(\phi \cdot \mathbf{f}) > 0 \quad \textit{for all} \quad \mathbf{x} \in D$$

or

$$\operatorname{div}(\phi \cdot \mathbf{f}) < 0 \quad \textit{for all} \quad \mathbf{x} \in D$$

holds, then no periodic orbits lie entirely in D*.*

Remark 2 ([80])**.** *The conditions stated in Theorem 8.1 can be relaxed to allow for* $\operatorname{div}(\phi \cdot \mathbf{f})$ *to have a null value only on null measure set, such as a single point in* $\mathbb{R}^2$*.*

To study the existence of periodic orbits in higher dimensions, Muldowney in [81] gave an extension of the classical BD criterion presented in Theorem 8.1 which can be applied to check the non-existence of closed orbits in simply connected sets in arbitrary dimension.

First, let us go over some relevant concepts and definitions to allow us to understand Muldowney's extension of the BD criterion.

Definition 8.1 (Multiplicative compound of a matrix [82]). *The k-th multiplicative compound $\mathbf{X}^{(k)}$ of $\mathbf{X}_{n\times m}$ is the $\binom{n}{k}\times\binom{m}{k}$ matrix that includes all the $k\times k$ minors of $\mathbf{X}$ written in lexicographic order, $x_{i_1\dots i_k}^{j_1\cdots j_k}$.*

Definition 8.2 (Additive compound of a matrix [81]). *The k-th additive compound $\mathbf{X}^{[k]}$ of the quadratic matrix $\mathbf{X}_{n\times n}$ is the $\binom{n}{k}\times\binom{n}{k}$ matrix defined by*

$$\mathbf{X}^{[k]} = D(I + h\mathbf{X})^{(k)}|_{h=0}$$

where D denotes differentiation with respect to h.

Remark 3. *In case $k = 1$ and $k = n$, we get $\mathbf{X}^{[1]} = \mathbf{X}, \quad \mathbf{X}^{[n]} = \operatorname{trace}(\mathbf{X})$, respectively.*

Definition 8.3 (Lozinskii norm [83]). *The Lozinskii norm $\mu(\cdot)$ of a square matrix $\mathbf{A}$ is defined as*

$$\mu(\mathbf{A}) = D_+|\mathbf{I} + h\mathbf{A}|_{h=0}$$

where D_+ denotes the right-hand derivative with respect to h.

The Lozinskii norm also known as logarithm norm depends on the matrix norm $|\cdot|$ chosen in its definition, the typical matrix norms used are 1-norm, ∞-norm and 2-norm.

The proposed extension of BD criterion develop by Muldowney is presented in Theorem 8.2 which gives a condition for global nonexistence of periodic orbits.

Theorem 8.2 (Muldowney's theorem [81]). *Consider system (2.1) with $\mathbf{x}\in\mathbb{R}^n$. If for all $\mathbf{x}\in\mathbb{R}^n$*

$$\mu\left(\frac{\partial \mathbf{f}^{[2]}}{\partial \mathbf{x}}\right) < 0 \quad \textit{or} \quad \mu\left(-\frac{\partial \mathbf{f}^{[2]}}{\partial \mathbf{x}}\right) < 0$$

holds, then system (2.1) has no non-constant periodic solutions.

Additionally, Muldowney presented a particular result for the nonexistence of periodic orbits on simply connected sets as shown in Lemma 5.

Lemma 5 ([81]). *Consider system (2.1), suppose $E\subseteq\mathbb{R}^n$ is a bounded open set and is invariant with respect to system 2.1. If for all $\mathbf{x}\in E$*

$$\mu\left(\frac{\partial \mathbf{f}^{[2]}}{\partial \mathbf{x}}\right) < 0 \quad \textit{or} \quad \mu\left(-\frac{\partial \mathbf{f}^{[2]}}{\partial \mathbf{x}}\right) < 0 \tag{8.1}$$

holds, then there is no periodic orbits lying entirely in E.

According to [81], if the underlying matrix norm in the Lozinskii norm $\mu(\cdot)$ is the 2-norm, then the condition shown in Theorem 8.2 can be written as follows,

$$\mu\left(\frac{\partial \mathbf{f}^{[2]}}{\partial \mathbf{x}}\right) = \lambda_1 + \lambda_2$$

where λ_1 and λ_2 are the largest eigenvalues of

$$\mathbf{F}(\mathbf{x}) = \frac{1}{2}\left[\mathbf{J}^\top(\mathbf{x}) + \mathbf{J}(\mathbf{x})\right], \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$$

with $\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$.

With the prior results, Smith in [84] has shown that for any dimension, the extended BD criterion presented by Muldowney can be used to check the existence of periodic orbits in continuously differentiable autonomous systems, but also that it implies that all bounded trajectories tend to an equilibrium point as time goes to infinity if the system is continuously differentiable.

The following theorem states that for any stable system, that is, for any system with bounded trajectories, all trajectories converge to an equilibrium point if a condition involving the truncated trace of $\mathbf{F}(\mathbf{x})$ holds for all space.

Theorem 8.3 (Theorem 7 from [84]). *Consider system (2.1) with $\mathbf{x} \in \mathbb{R}^n$. Let $\lambda_1(\mathbf{x})$ and $\lambda_2(\mathbf{x})$ be the largest eigenvalues of $\mathbf{F}(\mathbf{x})$. If for all $\mathbf{x} \in \mathbb{R}^n$*

$$\lambda_1(\mathbf{x}) + \lambda_2(\mathbf{x}) < 0 \tag{8.2}$$

holds, then each bounded trajectory converges to an equilibrium point.

Theorem 8.3 shows that for a system with a single equilibrium point, global asymptotic stability can be checked by evaluating a truncated trace of $\mathbf{F}(\mathbf{x})$, in this case the truncated trace involves only the two largest eigenvalues of $\mathbf{F}(\mathbf{x})$. Even though this theorem is powerful, because it guarantees asymptotic stability and, by indirectly denying the existence of other types of limits such as limit cycles, most nonlinear systems do not have asymptotic stability as a global property, and so this theorem falls short when considering nonlinear systems.

In Theorem 8.4, an extension of Theorem 8.3 is presented which allows for a local characterization of asymptotic stability, that is, it is possible to check if an invariant set is a subset of the true ROA by evaluating a truncated trace of matrix $\mathbf{F}(\mathbf{x})$.

Theorem 8.4 (Theorem 8 from [84]). *Suppose that D is a bounded simply-connected open subset of $\mathbb{R}^n$ with closure $\bar{D} \subset \mathbb{R}^n$. Suppose also that its boundary ∂D is crossed strictly inwards by every solution of system (2.1) which meets it. If for all $\mathbf{x} \in \bar{D}$*

$$\lambda_1(\mathbf{x}) + \lambda_2(\mathbf{x}) < 0 \tag{8.3}$$

holds, then each trajectory of system (2.1) in D converges to an equilibrium point.

Dealing with eigenvalues is a laborious task if we need to check the value of eigenvalues over a region in the state space. In this particular case, we want a specific truncated trace characterized by being the sum of the two largest eigenvalues of $\mathbf{F}(\mathbf{x})$. When the dynamical system has dimension two, Theorem 8.4 is the same as the BD-criterion if the function ϕ is equal to one, and we only consider the negative inequality, but also in this case the truncated trace is the sum of all elements of the diagonal of $\mathbf{F}(\mathbf{x})$.

Theorem 8.5 shows an equivalence condition to the condition shown in Theorem 8.4, avoiding the calculation of eigenvalues, and therefore the truncated trace to check convergence of each trajectory to an equilibrium point.

Theorem 8.5 ([84])**.** *Suppose that D is a bounded simply-connected open subset of $\mathbb{R}^n$ with closure $\bar{D} \subset \mathbb{R}^n$. Suppose also that its boundary ∂D is crossed strictly inwards by every solution of system (2.1). If there exist a constant real symmetric positive definite matrix $\mathbf{Q}$ and a continuous scalar function $\theta(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ such that for all $\mathbf{x} \in \bar{D}$ with $\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$*

$$\begin{aligned} &\mathbf{J}(\mathbf{x})^\top \mathbf{Q} + \mathbf{Q}\mathbf{J}(\mathbf{x}) + 2\theta(\mathbf{x})\mathbf{Q} \succeq 0 \\ &(n-2)\theta(\mathbf{x}) + \operatorname{tr}(\mathbf{J}(\mathbf{x})) < 0 \end{aligned} \tag{8.4}$$

holds, then each trajectory in D converges to an equilibrium point.

An important remark presented in [81] states that system (2.1) can be modified to be a system with different dynamics but with the same orbits by using a positive continuously differentiable scalar function. Consider system (2.1) with $\mathbf{x} \in \mathbb{R}^n$ and let $\phi(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, $\phi \in \mathcal{C}^1$, but allow $\phi(\mathbf{x})$ to be zero only at points which satisfy $\mathbf{f}(\mathbf{x}) = 0$. The autonomous system

$$\dot{\mathbf{x}} = \phi(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) \tag{8.5}$$

is denoted as an extension of system (2.1). This new system has different dynamics compared to the original system but has the same ROA and equilibrium points. Using this system in the iterative bilinear sum-of-squares algorithms to estimate the ROA and considering $\phi(\mathbf{x})$ as a polynomial decision variable, can allow for better estimates of the ROA.

For ease of notation, let's denote $\mathbf{J}_\phi$ as the jacobian of the extended system (8.5), i.e.,

$$\mathbf{J}_\phi(\mathbf{x}) = \left(\frac{\partial(\phi(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}))}{\partial \mathbf{x}} \right) \tag{8.6}$$

and denote $\mathbf{F}_\phi$ as $\frac{1}{2}(\mathbf{J}_\phi^\top(\mathbf{x}) + \mathbf{J}_\phi(\mathbf{x}))$ with $\mathbf{x} \in \mathbb{R}^n$.

Once again, just like introduced in Chapter 4, we will only deal with polynomial systems granting us the ability to check set containment conditions through the usage of the P-satz on the set containment conditions presented in the aforementioned theorems. Unlike the Lyapunov theory presented in Chapter 2, in Theorem 8.5 the conditions established to guarantee asymptotic stability, involve not only positiveness of scalar functions but also positiveness of symmetric polynomial matrices.

To understand how it is possible to check positiveness of symmetric polynomial matrices either globally or on a semi-algebraic set, we present the concept of polynomial matrix sum-of-squares and a theorem similar to P-satz that checks positive definiteness of polynomial matrices over a semi-algebraic set.

Definition 8.4. *Let $M(\mathbb{R}[\mathbf{x}])$ be the set of squared matrices of size n with entries belonging to $\mathbb{R}[\mathbf{x}]$. The set of all finite sums of elements of the form $\mathbf{A}(\mathbf{x})^\top \mathbf{A}(\mathbf{x})$ with $\mathbf{A}(\mathbf{x}) \in M(\mathbb{R}[\mathbf{x}])$, is denoted as the set of polynomial sum-of-squares matrices, $\Sigma M(\mathbb{R}[\mathbf{x}])^2$.*

Similarly to SOS polynomials, any symmetric polynomial matrix $\mathbf{A}(\mathbf{x})$ that is a sum-of-squares matrix is by definition positive semi-definite for all values of $\mathbf{x} \in \mathbb{R}^n$. The following lemma presents a SOS decomposition of a symmetric polynomial matrix.

Lemma 6 ([85])**.** *Any polynomial matrix $\mathbf{S}(\mathbf{x})$ of dimension p belongs to $\Sigma M(\mathbb{R}[\mathbf{x}])^2$ if and only if there exists a constant symmetric matrix $\mathbf{Z}$ such that*

$$\mathbf{S}(\mathbf{x}) = (\mathbf{u}(\mathbf{x}) \otimes \mathbf{I}_p)^\top \mathbf{Z} (\mathbf{u}(\mathbf{x}) \otimes \mathbf{I}_p)), \quad \mathbf{Z} \succeq 0 \tag{8.7}$$

with $\mathbf{u}(\mathbf{x})$ as a vector of monomials of $\mathbf{x}$. Here $\otimes$ denotes the Kronecker product of two matrices.

With Lemma 6, for a symmetric polynomial matrix $\mathbf{S}(\mathbf{x}) \in M(\mathbb{R}[\mathbf{x}])$, verifying if it is a SOS polynomial matrix simply consists in searching for the existence of some matrix $\mathbf{Z}$ according to (8.7). Ultimately, checking if a symmetric polynomial matrix $\mathbf{S}(\mathbf{x})$ is a SOS polynomial matrix consists in a standard LMI problem and consequently can be solved efficiently in polynomial time.

In [85], it is presented a theorem similar to the P-satz but for polynomial symmetric matrices. A symmetric polynomial matrix is positive definite over a compact semi-algebraic set if the condition in the following theorem is verified.

Theorem 8.6 (Theorem 2 from [85])**.** *If the symmetric-valued polynomial matrix $\mathbf{H}(\mathbf{x})$ of dimension p is positive definite on the compact set $G = \bigcap_{j=1}^{q} L(-g_j, 0)$ with $g_j \in \mathbb{R}[\mathbf{x}]$ for $j = 1, \ldots, q$, there exist $\epsilon > 0$ and sum-of-squares polynomials matrices $\mathbf{C}_0(\mathbf{x}), \ldots, \mathbf{C}_q(\mathbf{x})$ of dimension p such that*

$$\mathbf{H}(\mathbf{x}) = \mathbf{C}_0(\mathbf{x}) + \sum_{j=1}^{q} \mathbf{C}_j(\mathbf{x}) g_j(\mathbf{x}) + \epsilon \mathbf{I}_p \tag{8.8}$$

For the particular case of a symmetric polynomial matrix being positive definite in a semi-algebraic set described by a single polynomial inequality, Theorem 8.6 is specified by the following lemma.

Corollary 8.1. *Let K_g be the compact set defined by $K_g = L(-g, 0)$ with $g \in \mathbb{R}[\mathbf{x}]$. If a symmetric polynomial matrix $\mathbf{H}(\mathbf{x})$ of dimension p is positive definite on K_g, there exist $\epsilon > 0$ and sum-of-squares polynomial matrices $\mathbf{C}_0(\mathbf{x})$, $\mathbf{C}_1(\mathbf{x})$ such that*

$$\mathbf{H}(\mathbf{x}) = \mathbf{C}_0(\mathbf{x}) + \mathbf{C}_1(\mathbf{x}) g(\mathbf{x}) + \epsilon \mathbf{I}_p \tag{8.9}$$

A similar result is possible for positive semi-definite matrices, where for any symmetric polynomial matrix $\mathbf{H}(\mathbf{x})$ that can be decomposed into

$$\mathbf{H}(\mathbf{x}) = \mathbf{C}_0(\mathbf{x}) + \mathbf{C}_1(\mathbf{x})g(\mathbf{x}) \tag{8.10}$$

with $\mathbf{C}_0, \mathbf{C}_1 \in \Sigma M(\mathbb{R}[\mathbf{x}])^2$, then $\mathbf{H}(\mathbf{x})$ is inherently positive semi-definite on K_g.

Unlike positiveness of polynomial matrices, there is not an equivalency between non-negative matrices and the decomposition shown in equation (8.10), that is, there is no theorem that states that all non-negative matrices over a semi-algebraic set has a SOS decomposition, by finding polynomial SOS matrices, $\mathbf{C}_0$ and $\mathbf{C}_1$.

8.2 Computation of an estimate of the ROA

The iterative bilinear sum-of-squares algorithms proposed in this chapter, estimate a subset of the ROA similarly to the methods presented in Chapter 5, but instead of relying on a Lyapunov function to validate that an invariant set is a subset of the ROA associated with an equilibrium, it relies on the BD criterion and its extensions to higher dimensions. There is no need for a Lyapunov function, but we rely on the extended system described in (8.5) which has a function that will be a decision variable in the proposed methods. This function works similarly to a Lyapunov function because its choice has an influence on the subset of the ROA obtained. Further explanation on the development of the proposed algorithms and the procedure taken is presented below.

8.2.1 BD approach for planar dynamics

To use the theorems presented before, we need to impose a condition to ensure that the set we are working on is an invariant set with respect to system (2.1), so that all trajectories generated inside the set remain in the set. For it let us consider a polynomial function $R \in \mathbb{R}[\mathbf{x}]$ and a positive scalar γ such that the following set containment condition holds,

$$\partial L(R, \gamma) \subseteq L^{\circ}(\nabla R \cdot \mathbf{f}, 0) \tag{8.11}$$

For all values of γ which the set containment condition shown in (8.11) holds, it guarantees that the set $\partial L(R, \gamma)$ is an invariant through the use of Nagumo's Theorem shown in Chapter 2. Applying Lemma 4 to this set containment condition, gives that if

$$\begin{aligned} &s_2(R - \gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ &s_2 \in \mathbb{R}[\mathbf{x}] \end{aligned} \tag{8.12}$$

holds, then the set containment condition described in equation (8.11) is verified. As a remark, again as seen in Chapter 5, the function $l(\mathbf{x})$ present in the SOS conditions is an auxiliary function to guarantee strict positivity or negativity.

Even though, we have a condition to guarantee strict invariance of a set, we still need to ensure that all trajectories generated inside that set converge to the equilibrium point inside and no limit cycle or periodic orbits exist. Let's first look at the case of planar dynamics where Lemma 5 can be applied as follows to the strict invariant set $L(R,\gamma)$ resulting in

$$L(R,\gamma) \subseteq L^{\circ}(\mathrm{div}(\phi \cdot \mathbf{f}), 0) \tag{8.13}$$

Equation (8.13) resorts to an additional continuously differentiable polynomial function $\phi(\mathbf{x})$, that can be interpreted as a Lyapunov function because its relation with the dynamics imposes conditions on the ROA estimate, that is, for different choices of $\phi(\mathbf{x})$ the resulted ROA estimate is different.

Again, let's apply Lemma 4 to the set containment condition described in equation (8.13), and we get the SOS conditions shown in equation (8.14),

$$\begin{aligned} &s_1(R-\gamma) - \mathrm{div}(\phi \cdot \mathbf{f}) - l \in \Sigma[\mathbf{x}] \\ &s_1 \in \Sigma[\mathbf{x}] \end{aligned} \tag{8.14}$$

Combining the SOS conditions described in equations (8.12) and (8.14) we get the following set of SOS conditions that guarantees that a set is invariant and that it is a subset of the ROA of system (2.1).

$$\begin{aligned} &s_1(R-\gamma) - \mathrm{div}(\phi \cdot \mathbf{f}) - l \in \Sigma[\mathbf{x}] \\ &s_1 \in \Sigma[\mathbf{x}] \\ &s_2(R-\gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ &s_2 \in \mathbb{R}[\mathbf{x}] \end{aligned} \tag{8.15}$$

Similar to the algorithms presented in Chapter 5, we intend to convert these conditions into an iterative SOS algorithm that converges as close as possible to the true ROA. To ensure convergence to the true ROA, that is, to ensure enlargement of the estimate of the region of attract in each iteration, an additional SOS condition is necessary. First, let's define the set containment condition used to ensure enlargement of the estimate over each iteration by

$$L(\bar{R},\bar{\gamma}) \subseteq L(R,\gamma) \tag{8.16}$$

with $L(\bar{R},\bar{\gamma})$ as the previous estimate of the ROA and $L(R,\gamma)$ as the new estimate that needs to contain the previous estimate. Again, let's apply Lemma 4 to convert the enlargement set condition into a SOS conditions as follows,

$$\begin{aligned} &(\gamma - R) - s_3(\bar{\gamma} - \bar{R}) \in \Sigma[\mathbf{x}] \\ &s_3 \in \Sigma[\mathbf{x}] \end{aligned} \tag{8.17}$$

Now that we have all the necessary SOS conditions, Algorithm 6, labelled as BD-planar presents a possible implementation, where the inputs are an initial guess of the function $R(\mathbf{x})$ and a constant function $l(\mathbf{x})$ for strict positivity purposes as explained before. It is a two-step iterative bilinear sum-of-squares algorithm, with the first step in charge of calculating feasible s-multipliers and the second step to obtain a new estimate of the ROA.

Algorithm 6. *BD-approach for planar systems* *(BD-planar)*

Input: *polynomial* R, $l = \epsilon(\mathbf{x}^\top \mathbf{x})$ *with* $\epsilon = 10^{-6}$

Step I: *solve for* s_1, s_2, ϕ *and* γ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & s_1(R-\gamma) - \operatorname{div}(\phi \cdot \mathbf{f}) - l \in \Sigma[\mathbf{x}],\ s_1 \in \Sigma[\mathbf{x}] \\ & s_2(R-\gamma) - (\nabla R)\cdot \mathbf{f} - l \in \Sigma[\mathbf{x}],\ s_2 \in \mathbb{R}[\mathbf{x}] \\ & \phi - l \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(R,\gamma) \subseteq L^{\circ}(\operatorname{div}(\phi \cdot \mathbf{f}), 0) \\ & \partial L(R,\gamma) \subseteq L^{\circ}(\nabla R \cdot \mathbf{f}, 0) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad \phi(\mathbf{x}) > 0 \end{aligned}$$

Step II: *solve for* R *and* ϕ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & s_1(R-\gamma) - \operatorname{div}(\phi \cdot \mathbf{f}) - l \in \Sigma[\mathbf{x}] \\ & s_2(R-\gamma) - (\nabla R)\cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ & s_3(\bar{R}-\bar{\gamma}) - (R-\gamma) \in \Sigma[\mathbf{x}],\ s_3 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(R,\gamma) \subseteq L^{\circ}(\operatorname{div}(\phi \cdot \mathbf{f}), 0) \\ & \partial L(R,\gamma) \subseteq L^{\circ}(\nabla R \cdot \mathbf{f}, 0) \\ & L(\bar{R},\bar{\gamma}) \subseteq L(R,\gamma) \end{aligned}$$

$\bar{R}$ *and* $\bar{\gamma}$ *denote the solution obtained for* R *and* γ *in Step I*

Step III: *Go to Step I unless maximum number of iterations is reached or when termination condition is met.*

Output: R *and* γ

Let's try to get better insight into what it is really happening in this algorithm and how does the degree of the auxiliary function $\phi(\mathbf{x})$, affects the estimate of the ROA. First let's consider the case when $\phi(\mathbf{x})$ is chosen to be constant, in particular we chose $\phi = 1$ without loss of generality because different values of the constant only rescale the divergence of the system and does not change the location of the zero level set of $\operatorname{div}(\mathbf{f})$. Figure 8.1 shows the results of applying BD-planar to system (S1) with $\phi(\mathbf{x})$ as a constant.

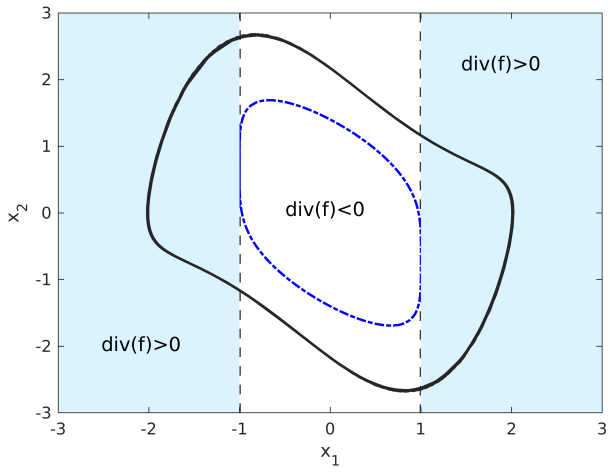


Figure 8.1: Results from applying BD-planar to system (S1) with $\phi(\mathbf{x})$ of degree 0. The blue line (-·-·-) represents the subset of the ROA obtained, and the black line (——) represents the true ROA.

We can see with Figure 8.1 that the invariant set is being constrained by the zero level set of the divergence of the system. With $\phi(\mathbf{x})$ as a constant, it grants the algorithm with the problem of only finding an invariant set constrained by the zero level set of the divergence which from the results shown, poor results arise.

Now let's consider the function $\phi(\mathbf{x})$ as a polynomial decision variable of degree six in BD-planar and the results from applying it to system (S1) is presented in Figure 8.2.

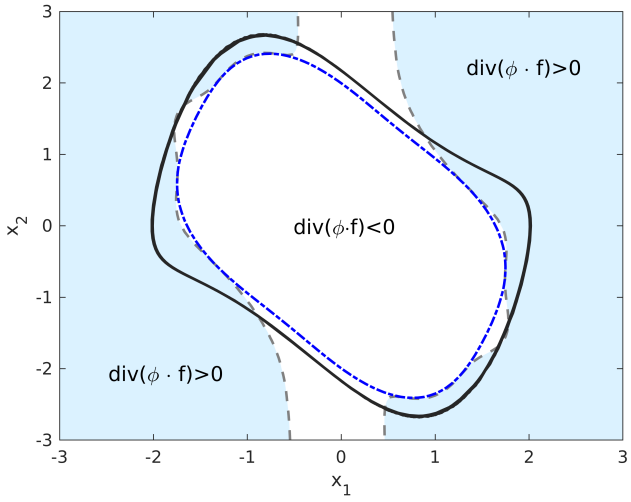


Figure 8.2: Subset of the ROA by applying BD-planar to system (S1) with $\phi(\mathbf{x})$ of degree six. The blue line (-·-·-) represents the subset of the ROA obtained, and the black line (——) represents the true ROA.

The use of the function $\phi(\mathbf{x})$ as a polynomial decision variable of degree 6 allows to morph the zero level set of the divergence of the original dynamics and as seen in Figure 8.2, as it provides a better estimate of the ROA. To check the effect of different choices of the degree of $\phi(\mathbf{x})$, some runs were performed on system (S1) for a range of different degrees of $\phi(\mathbf{x})$. Figure 8.3 shows results for different degrees of $\phi(\mathbf{x})$ and for the case of $\phi(\mathbf{x}) = 1$.

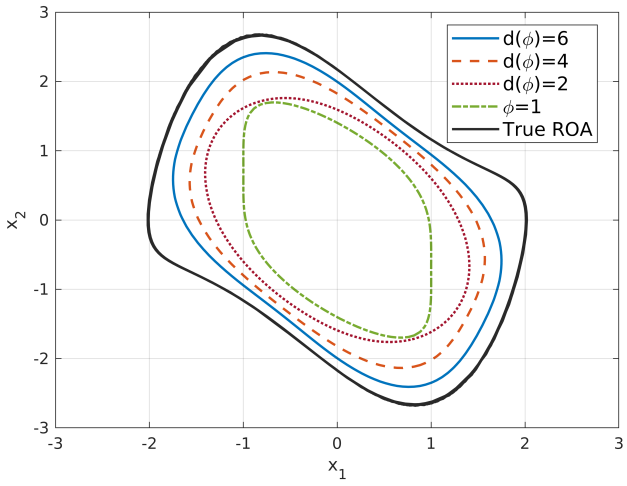
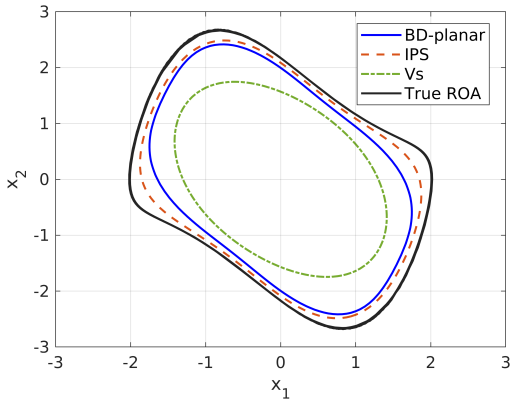


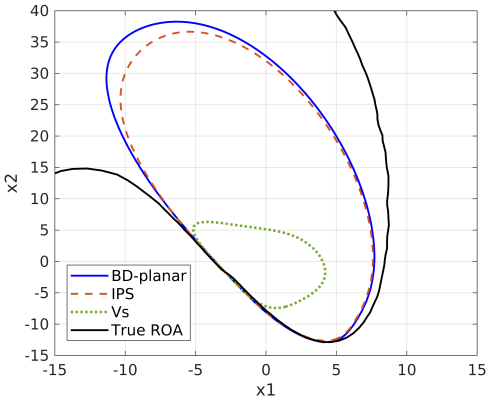
Figure 8.3: Subsets of the ROA after applying BD-planar to system (S1) with different degrees of $\phi(\mathbf{x})$.

From the results in Figure 8.3, we observe that in general, a higher choice of the degree of $\phi(\mathbf{x})$ improves the estimate of the ROA because a higher degree polynomial is able to distort the zero level set of the divergence of the original system such that a bigger estimate is reached. However, applying higher degrees of $\phi(\mathbf{x})$ comes with a higher computation cost due to the increase number of decision variables and due to the growing size of the LMI in each step of the algorithm.

To compare BD-planar to the current algorithms in the literature, we applied Vs and IPS to systems (S1) and (S3) alongside the current proposal based on BD criterion for planar dynamics. The IPS is quite similar to the proposed algorithm, at least with respect to the condition used to enlarge the estimate. The results are presented in Figures 8.4(a) and 8.4(b) considering the subset of the ROA estimated by each algorithm described by a polynomial of degree four.



(a) Subsets of the ROA obtained from applying algorithms Vs, IPS and BD-planar to system (S1).



(b) Subsets of the ROA obtained from applying algorithms Vs, IPS and BD-planar to system (S3).

Figure 8.4: Application of BD-planar to two case studies and comparative results.

From the results in Figure 8.4, the BD-planar has similar results to the IPS algorithm as expected, mostly due to both having the same enlargement condition imposed in the algorithm. For system (S1), it shows that the BD-planar algorithm has a slight worse result compared to the IPS and in contrast, for the system (S3), it has a better performance; it is relevant to point out that the IPS algorithm stopped due to feasibility issues unlike the BD-planar which stopped due to it converging to a value.

From experience and tests in multiple systems, the BD-planar does not have a better performance in systems with a bounded ROA, but when we consider systems with an unbounded ROA, such as system (S3), sometimes it has better results in comparison with the IPS and Vs algorithms.

The previous algorithm based on the BD criterion on planar systems falls short because it is inherently only applicable to planar dynamics and in general most systems in real life are not only described in $\mathbb{R}^2$. Two algorithms based on extensions of the BD criterion introduced at the beginning of this chapter will be presented in the remaining portion of this section. To create and develop those algorithms, we turn to the notions of SOS matrices and definite positiveness over a compact semi-algebraic set that were introduced at the beginning of this chapter.

8.2.2 BD approach for higher dimensions – Method 1

The first proposed algorithm based on BD extensions uses Theorem 8.4, which states that if a truncated trace is negative definite over an invariant set then it is a subset of the ROA. The issue of this is that it is a truncated trace and for dimensions bigger than two, we need to calculate the eigenvalues for any $\mathbf{x}$ in the considered invariant set to check the condition. The workaround proposed is to instead of checking if condition (8.3) holds, the condition is relaxed such that we guarantee that the polynomial matrix $\mathbf{F}$ is negative definite, hence the condition (8.3) is always met, although with some conservatism. We resort to the extended system (8.5) to enable better results as it is used to tackle the conservatism imposed by the relaxed condition, the extended system works similarly to the use of function $\phi(\mathbf{x})$ in the BD-planar. The procedure taken is similar to the one taken in BD-planar, where set containment conditions are converted into SOS conditions, the only difference is the condition employed to guarantee asymptotic stability which in this case is by checking the positivity of a polynomial matrix. In other words we enforce the invariant set $L(R,\gamma)$ with $R \in \mathbb{R}[\mathbf{x}]$ and $\gamma \in \mathbb{R}$ to be contained inside the set where matrix $\mathbf{F}_\phi(\mathbf{x})$ with $\phi(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, is negative definite, i.e.,

$$\begin{aligned} &L(R,\gamma) \subseteq L^\circ(\mathbf{J}_\phi^\top + \mathbf{J}_\phi, 0) \\ &\forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad \phi(\mathbf{x}) > 0 \end{aligned} \tag{8.18}$$

The set containment condition involving matrix positivity on a semi-algebraic set shown in (8.18) can be converted into a SOS condition by using Lemma 8.1, which certifies condition (8.18) if there exist a polynomial SOS matrix $\mathbf{C}_1 \in \Sigma M(\mathbb{R}[\mathbf{x}])^2$ and a scalar $\epsilon > 0$ such that the following condition holds.

$$-\mathbf{J}_\phi^\top - \mathbf{J}_\phi - \mathbf{C}_1(\gamma - R) - \epsilon \mathbf{I} \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \tag{8.19}$$

The first proposed extension of BD-planar to higher dimensions is presented in Algorithm 7, labeled BD-1.

Algorithm 7. *Extended BD-approach (1º method)* *(BD-1)*

Input: *polynomial* R, $l = \epsilon(\mathbf{x}^\top \mathbf{x})$ *with* $\epsilon = 10^{-6}$

Step I: *hold* R *and solve for* $\mathbf{C}_1$, ϕ, s_1 *and* γ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -\mathbf{J}_\phi^\top - \mathbf{J}_\phi - \mathbf{C}_1(\gamma - R) - \epsilon\mathbf{I} \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\ & s_1(R-\gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}],\ s_1 \in \mathbb{R}[\mathbf{x}] \\ & \phi - l \in \Sigma[\mathbf{x}],\ \mathbf{C}_1 \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(R,\gamma) \subseteq L^\circ(\mathbf{J}_\phi^\top + \mathbf{J}_\phi, 0) \\ & \partial L(R,\gamma) \subseteq L^\circ(\nabla R \cdot \mathbf{f}, 0) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad \phi(\mathbf{x}) > 0 \end{aligned}$$

Step II: *hold* $\mathbf{C}_1$, s_1, ϕ *and solve for* R, s_2 *and* γ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & -\mathbf{J}_\phi^\top - \mathbf{J}_\phi - \mathbf{C}_1(\gamma - R) - \epsilon\mathbf{I} \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\ & s_1(R-\gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ & -(\bar{\gamma} - \bar{R})s_2 + \gamma - R \in \Sigma[\mathbf{x}],\ s_2 \in \Sigma[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(R,\gamma) \subseteq L^\circ(\mathbf{J}_\phi^\top + \mathbf{J}_\phi, 0) \\ & \partial L(R,\gamma) \subseteq L^\circ(\nabla R \cdot \mathbf{f}, 0) \\ & L(\bar{R},\bar{\gamma}) \subseteq L(R,\gamma) \end{aligned}$$

$\bar{R}$ *and* $\bar{\gamma}$ *denote the solution obtained for* R *and* γ *in Step I*

Step III: *Go to Step I except when maximum number of iterations is reached or when termination condition is met.*

Output: R *and* γ

Due to the condition imposed in the development of BD-1 that the subset of the ROA needs to be contained in the region where all eigenvalues of $\mathbf{F}_\phi(\mathbf{x})$ are negative, from some tests in multiple high dimensional systems, it revealed to be quite conservative and in most cases the resulted estimate is poor, and sometimes it is not even possible to obtain an estimate even for high degree of $\phi(\mathbf{x})$. In other words, the relaxation done in this algorithm made it highly system dependent and because of that it is hard to guarantee that it will work for any given polynomial dynamical system.

8.2.3 BD approach for higher dimensions – Method 2

The second algorithm developed based on BD extensions is presented in Algorithm 8, labeled as BD-2, and it relies on Theorem 8.5 which does not rely on checking a truncated trace, instead two equivalent conditions are stated. The first condition consists of checking if a matrix is positive semidefinite over an invariant set and the second is of checking if a scalar function is negative definite over the same invariant set.

We guarantee that the invariant set $D = L(R, \gamma)$ with respect to system (2.1) is a subset of the ROA if there exist a symmetric positive matrix $\mathbf{Q}$ and a scalar function $\theta(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ such that for all $\mathbf{x} \in D$

$$\begin{aligned} &\mathbf{J}^\top(\mathbf{x})\mathbf{Q} + \mathbf{Q}\mathbf{J}(\mathbf{x}) + 2\theta(\mathbf{x})\mathbf{Q} \succeq 0 \\ &(n-1)\theta(\mathbf{x}) + \operatorname{tr}(\mathbf{J}(\mathbf{x})) < 0 \end{aligned} \tag{8.20}$$

holds. Unlike BD-1, in BD-2 there is no need to relax the condition with regard to the eigenvalues of a matrix, but it has the issue of having bilinear decision variables, more specifically, the scalar function $\theta(\mathbf{x})$ and the constant matrix $\mathbf{Q}$. The workaround used in this case was to consider the matrix $\mathbf{Q}$ as a given constant matrix, in the case presented we considered it as the identity matrix, resulting in the following conditions that should hold for all $\mathbf{x}$ in the invariant set $L(R, \gamma)$, with $R \in \mathbb{R}[\mathbf{x}]$ and $\gamma \in \mathbb{R}$.

$$\begin{aligned} &\mathbf{J}^\top(\mathbf{x}) + \mathbf{J}(\mathbf{x}) + 2\theta(\mathbf{x})\mathbf{I} \succeq 0 \\ &(n-1)\theta(\mathbf{x}) + \operatorname{tr}(\mathbf{J}(\mathbf{x})) < 0 \end{aligned} \tag{8.21}$$

As it is expected this may constrain the algorithm and result in poor estimates of the ROA, because we are narrowing the condition to guarantee asymptotic stability inside an invariant set.

Similarly to BD-1, we resort to the extended system (8.5) to tackle a little the relaxation used, resulting in the following set containment conditions to guarantee asymptotic stability in the invariant set $L(R, \gamma)$, with $R \in \mathbb{R}[\mathbf{x}]$ and $\gamma \in \mathbb{R}$.

$$\begin{aligned} &L(R,\gamma) \subseteq L^\circ(-\mathbf{J}_\phi^\top - \mathbf{J}_\phi - 2\theta\mathbf{I}, 0) \\ &L(R,\gamma) \subseteq L^\circ((n-2)\theta + \operatorname{tr}(\mathbf{J}_\phi), 0) \end{aligned} \tag{8.22}$$

Using Lemma 4 and 8.1, the set containment conditions can be converted into SOS conditions, as follows.

$$\begin{aligned} &\mathbf{J}_\phi^\top + \mathbf{J}_\phi + 2\theta\mathbf{I} - \mathbf{C}_1(\gamma - R) \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\ &-(\gamma - R)s_1 - (n-2)\theta - \operatorname{tr}(\mathbf{J}_\phi) - l \in \Sigma[\mathbf{x}] \end{aligned} \tag{8.23}$$

Algorithm 8. *Extended BD-approach (2º method)* *(BD-2)*

Input: *polynomial* R *and* $l = \epsilon(\mathbf{x}^\top\mathbf{x})$ *with* $\epsilon = 10^{-6}$

Step I: *hold* R *and solve for* $\mathbf{C}_1$, ϕ, s_1, s_2, θ *and* γ *such that*

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & \mathbf{J}_\phi^\top + \mathbf{J}_\phi + 2\theta\mathbf{I} - \mathbf{C}_1(\gamma - R) \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\ & -(\gamma - R)s_1 - (n-2)\theta - \operatorname{tr}(\mathbf{J}_\phi) - l \in \Sigma[\mathbf{x}] \\ & s_2(R - \gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\ & \phi - l \in \Sigma[\mathbf{x}],\ s_1 \in \Sigma[\mathbf{x}] \\ & \mathbf{C}_1 \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\ & s_2,\ \theta \in \mathbb{R}[\mathbf{x}] \end{aligned} \qquad \begin{aligned} \max \quad & \gamma \\ \text{s.t.} \quad & L(R,\gamma) \subseteq L^\circ(-\mathbf{J}_\phi^\top - \mathbf{J}_\phi - 2\theta\mathbf{I}, 0) \\ & L(R,\gamma) \subseteq L^\circ((n-2)\theta + \operatorname{tr}(\mathbf{J}_\phi), 0) \\ & \partial L(R,\gamma) \subseteq L^\circ(\nabla R \cdot \mathbf{f}, 0) \\ & \forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\}, \quad \phi(\mathbf{x}) > 0 \\ & \phi(0) \geq 0 \end{aligned}$$

Step II: *hold* $\mathbf{C}_1$, s_1, s_2, ϕ, θ *and solve for* R, s_3 *and* γ *such that*

$$
\begin{aligned}
\max \quad & \gamma \\
\text{s.t.} \quad & \mathbf{J}_\phi^\top + \mathbf{J}_\phi + 2\theta\mathbf{I} - \mathbf{C}_1(\gamma - R) \in \Sigma M(\mathbb{R}[\mathbf{x}])^2 \\
& -(\gamma - R)s_1 - (n-2)\theta - \operatorname{tr}(\mathbf{J}_\phi) - l \in \Sigma[\mathbf{x}] \\
& s_2(R-\gamma) - \nabla R \cdot \mathbf{f} - l \in \Sigma[\mathbf{x}] \\
& -(\bar{\gamma} - \bar{R})s_3 + \gamma - R \in \Sigma[\mathbf{x}] \\
& s_3 \in \Sigma[\mathbf{x}]
\end{aligned}
\qquad
\begin{aligned}
\max \quad & \gamma \\
\text{s.t.} \quad & L(R,\gamma) \subseteq L^\circ(-\mathbf{J}_\phi^\top - \mathbf{J}_\phi - 2\theta\mathbf{I}, 0) \\
& L(R,\gamma) \subseteq L^\circ((n-2)\theta + \operatorname{tr}(\mathbf{J}_\phi), 0) \\
& \partial L(R,\gamma) \subseteq L^\circ(\nabla R \cdot \mathbf{f}, 0) \\
& L(\bar{R},\bar{\gamma}) \subseteq L(R,\gamma)
\end{aligned}
$$

$\bar{R}$ *and* $\bar{\gamma}$ *denote the solution obtained for* R *and* γ *in Step I*

Step III: *Go to Step I except when maximum number of iterations is reached or when termination condition is met.*

Output: R *and* γ

To illustrate the performance of BD-1 and BD-2, we considered two examples, system (S7) and system (S8) characterized in $\mathbb{R}^3$ previously presented in Chapter 7.

For the system (S8), it revealed quite complicated because neither of these two algorithms got a feasible solution even for different choices of degrees of the decision variables in each algorithm such as $\phi(\mathbf{x})$, the s-multipliers $s_i(\mathbf{x})$ and the polynomial that describes the semi-algebraic set $R(\mathbf{x})$. This is likely due to the simplifications done in both that disable to work on this system. That is, both algorithms may fail or even have poor results due to the simplifications done.

Results from applying the IPS, Vs, BD-2 and BD-1 algorithms to system (S7) are presented in Figure 8.5 considering a subset of the ROA characterized by a single polynomial of degree four.

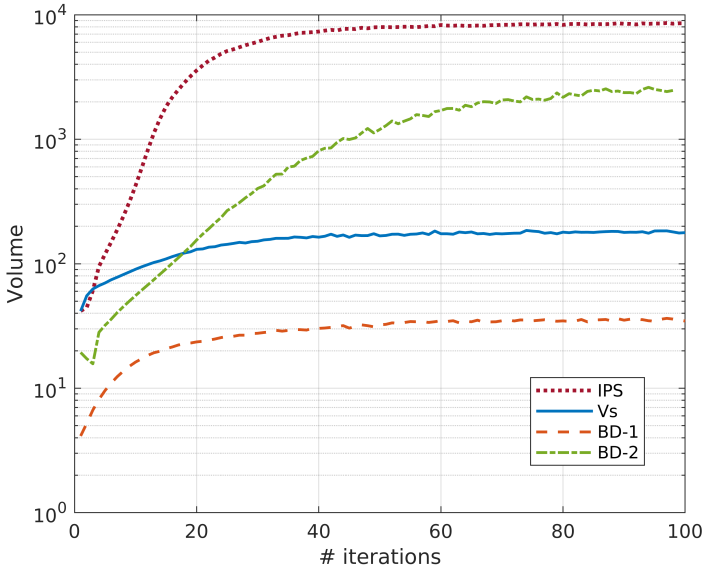


Figure 8.5: Comparative results between methods based on BD extensions (BD-1 and BD-2) and Lyapunov based methods (Vs and IPS).

From the results shown in Figure 8.5, it is clear that the BD-1 exhibits poor results, this is due to the conservatism imposed by enforcing all the eigenvalues of $\mathbf{F}(\mathbf{x})$ to be negative. By contrast, the BD-2 has good performance in relation to the volume of the estimate of the ROA obtained, although for this specific case study it performs worse than the IPS, likely because the condition imposed to be a subset of the ROA is stricter than in the Lyapunov based approach or because the simplification proposed also constrains the set of possible solutions of the problem and consequently a better estimate may not be possible to achieve.

Due to the use of SOS matrix conditions, which are more challenging to work with than simple polynomial SOS conditions, both proposed algorithms require more computation time per iteration than the algorithms provided in Chapter 5.

To sum up, the proposed algorithms based on Bendixson-Dulac criterion and extensions perform well on planar dynamics but have many issues in higher dimensional systems because of the simplifications and constraints bestowed upon them, leading to problems regarding feasibility and performance, that is, the estimate obtained is quite small in comparison with the real ROA and other well known algorithms, such as the IPS.

CHAPTER 9

Conclusions and Future Work

This chapter presents the main conclusions on the work described in the thesis, its contributions and discusses possible future work that can be a topic of more advanced research and development.

9.1 Conclusions

The presented work considered the problem of estimating the region of attraction of nonlinear systems described by polynomial vector fields. An overview of the methods currently in the literature are shown and discussed, pointing out their advantages and drawbacks. We primarily focused on iterative bilinear sum-of-squares region of attraction estimation algorithms due to their recent development and stability guarantees. The description of a few of these algorithms is presented alongside a new algorithm that consists in an enhancement of an already well-known algorithm called V-s iteration, by employing the use of two adaptive shape functions.

An analysis and comparison of these algorithms was performed on a proposed set of problems and procedures in order to obtain properties such as convergence, robustness, time of computation, scalability, likelihood of numerical issues and the presence of odd behaviors in the algorithms. We can conclude with the presented results that the IPS algorithm is systematically better in terms of accuracy of its estimation of the region of attraction compared to the other algorithms, but it has the drawback of being computationally more laborious and more likely to encounter computational issues. These unpleasant characteristics are intensified when dealing with higher dimensional systems, implicating that the IPS algorithm has more scalability issues compared to the other algorithms. The proposed enhancement, labeled Vsp-c, leads in general to a better accuracy of its estimate of the region of attraction and reveals to be more robust to initial conditions compared to its predecessor, but it comes with the drawback of relying on an additional optimization step which leads to an increase in computation time per iteration.

An iterative bilinear sum-of-squares algorithm to estimate the region of attraction of planar polynomial dynamical systems based on Bendixson-Dulac criterion and set invariance was presented. From a few studies, this algorithm has good performance, as it has similar results to the IPS algorithm.

Two methods to extend the aforementioned algorithm to higher dimensions were proposed, revealing both to be conservative and not reliable for a generic polynomial dynamical system of higher dimensions. This issue is due to the simplifications and restrictions used to develop these two iterative bilinear sum-of-squares algorithms.

9.2 Future Work

Future works can, on one hand, extend and further develop the set of problems and procedures which compose the proposed framework, focusing more on the dependence of initial conditions such as the s-multipliers degree and initial guesses of polynomial functions in order to establish a method of selecting these parameters that will guarantee a good performance of the algorithm with respect to time of computation, likelihood of computational issues and accuracy. On the other hand, the presented and explored idea of using invariant sets and Bendixson-Dulac criterion and its extensions to estimate the region of attraction by means of a bilinear sum-of-squares algorithm can be further studied as an alternative to Lyapunov-based methods, and they can be further explored in order to diminish the drawbacks the simplifications performed produced.

Bibliography

[1] U. Topcu and A. Packard, “Linearized analysis versus optimization-based nonlinear analysis for nonlinear systems,” *Proceedings of the American Control Conference*, pp. 790–795, 2009.

[2] L. R. Ray and R. F. Stengel, “A monte carlo approach to the analysis of control system robustness,” *Automatica*, vol. 29, no. 1, pp. 229–236, 1993.

[3] M. Heller, R. David, and J. Holmberg, “Falling Leaf Motion Suppression in the F/A-18 Hornet with Revised Flight Control Software,” *American Institute of Aeronautics and Astronautics*, 2004.

[4] M. Heller, R. J. Niewoehner, and K. P. Lawson, “F/A-18E/F Super Hornet High-Angle-of-Attack Control Law Development and Testing,” *Journal of Aircraft*, vol. 38, no. 5, pp. 841–847, 2001.

[5] M. Heller, R. David, and J. Holmberg, “Falling Leaf Motion Suppression in the F/A-18 Hornet with Revised Flight Control Software,” *AIAA Aerospace Sciences Meeting and Exhibit*, 2004. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2004-542

[6] T. Cunis, L. Burlion, and J. P. Condomines, “Piecewise polynomial modeling for control and analysis of aircraft dynamics beyond stall,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 4, pp. 949–957, 2019.

[7] A. Chakraborty, P. Seiler, and G. J. Balas, “Nonlinear region of attraction analysis for flight control verification and validation,” *Control Engineering Practice*, vol. 19, no. 4, pp. 335–345, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2010.12.001

[8] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, “Lyapunov-stable neural-network control,” 2021. [Online]. Available: https://arxiv.org/abs/2109.14152

[9] Y. Zhang, Y. Li, K. Tomsovic, S. M. Djouadi, and M. Yue, “Review on set-theoretic methods for safety verification and control of power system,” *IET Energy Systems Integration*, vol. 2, no. 3, pp. 226–234, 2020. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-esi.2019.0133

[10] T. Zhao, J. Wang, X. Lu, and Y. Du, “Neural Lyapunov Control for Power System Transient Stability: A Deep Learning-Based Approach,” *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 955–966, 2022.

[11] M. Vatani and M. Hovd, “Control of bilinear power converters using sum of squares programming,” *European Control Conference*, pp. 654–659, 2015.

[12] G. Chesi, “Estimating the domain of attraction for non-polynomial systems via LMI optimizations,” *Automatica*, vol. 45, no. 6, pp. 1536–1541, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2009.02.011

[13] X. Chen, J. Cao, J. H. Park, and J. Qiu, “Stability analysis and estimation of domain of attraction for the endemic equilibrium of an SEIQ epidemic model,” *Nonlinear Dynamics*, vol. 87, no. 2, pp. 975–985, 2016. [Online]. Available: https://doi.org/10.1007/s11071-016-3092-7

[14] R. Riah, M. Fiacchini, and M. Alamir, “Iterative method for estimating the robust domains of attraction of non-linear systems: Application to cancer chemotherapy model with parametric uncertainties,” *European Journal of Control*, vol. 47, pp. 64–73, 2019.

[15] F. Camilli, F. Wirth, and L. Grüne, “Calculating the domain of attraction : Zubov’s method and extensions,” *Proceedings of the Stability and Control Processes Conference*, pp. 27–36, 2005.

[16] F. Camilli, L. Grüne, and F. Wirth, “A regularization of Zubov’s equation for robust domains of attraction,” *Nonlinear control in the Year 2000*, pp. 277–289, 2007.

[17] F. Camilli and L. Grüne and F. Wirth, “Control lyapunov functions and Zubov’s method,” *SIAM Journal on Control and Optimization*, vol. 47, no. 1, pp. 301–326, 2008.

[18] V. I. Zubov, *Methods of A. M. Lyapunov and their Application.* P. Noordhoff, 1965.

[19] D. Gray, “Applications of Control Engineering,” *Centralized and Automatic Controls in Ships*, pp. 175–178, 1966.

[20] E. Najafi, R. Babuška, and G. A. Lopes, “A fast sampling method for estimating the domain of attraction,” *Nonlinear Dynamics*, vol. 86, no. 2, pp. 823–834, 2016.

[21] G. Yuan and Y. Li, “Estimation of the regions of attraction for autonomous nonlinear systems,” *Transactions of the Institute of Measurement and Control*, vol. 41, no. 1, pp. 97–106, 2019.

[22] A. El-Guindy, D. Han, and M. Althoff, “Estimating the region of attraction via forward reachable sets,” *Proceedings of the American Control Conference*, pp. 1263–1270, 2017.

[23] D. Henrion and M. Korda, “Convex computation of the region of attraction of polynomial control systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2014.

[24] F. Hamidi, H. Jerbi, W. Aggoune, M. Djemai, and M. Naceur Abdkrim, “Enlarging region of attraction via LMI-based approach and Genetic Algorithm,” *Int. Conference on Communications, Computing and Control Applications*, pp. 7–12, 2011.

[25] A. Prestel and C. N. Delzell, *Positive Polynomials: From Hilbert’s 17th Problem to Real Algebra*. Springer, 2001.

[26] A. A. Ahmadi, “Algebraic relaxations and hardness results in polynomial optimization and lyapunov analysis,” *arXiv*, 2012. [Online]. Available: https://arxiv.org/abs/1201.2892

[27] W. Tan, “Nonlinear Control Analysis and Synthesis using Sum-of-Squares Programming,” Ph.D. dissertation, University of California, Berkeley, 2006.

[28] S. Siegmund, C. Nowak, and J. Diblik, “A generalized picard-lindelöf theorem,” *Electronic Journal of Qualitative Theory of Differential Equations*, no. 28, 2016. [Online]. Available: https://doi.org/10.14232/ejqtde.2016.1.28

[29] C. S. Un and C. Lan-sun, *Theory of Limit Cycles*, 2nd ed. American Mathematical Society, 1986.

[30] M. Nagumo, “Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen,” *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[31] K. Ghorbal, A. Sogokon, and A. Platzer, “A hierarchy of proof rules for checking positive invariance of algebraic and semi-algebraic sets,” *International Conference on Verification, Model Checking, and Abstract Interpretation*, vol. 47, pp. 19–43, 2017.

[32] H. Khalil, *Nonlinear systems*, 3rd ed. Prentice Hall, 2007.

[33] A. Iannelli, A. Marcos, and M. Lowenberg, “Algorithms for the estimation of the region of attraction with positively invariant sets,” *International Conference on Systems and Control*, pp. 93–98, 2018.

[34] G. Datseris and A. Wagemakers, “Effortless estimation of basins of attraction,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 32, no. 2, p. 23104, 2022. [Online]. Available: https://doi.org/10.1063/5.0076568

[35] R. Genesio, M. Tartaglia, and A. Vicino, “On the Estimation of Asymptotic Stability Regions: State of the Art and New Proposals,” *IEEE Trans. Autom. Control*, vol. 30, no. 8, pp. 747–755, 1985.

[36] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[37] R. Baier and M. Gerdts, “A computational method for non-convex reachable sets using optimal control,” *European Control Conference*, pp. 97–102, 2009.

[38] A. Majumdar, R. Vasudevan, M. M. Tobenkin, and R. Tedrake, “Convex Optimization of Nonlinear Feedback Controllers via Occupation Measures,” *The International Journal of Robotics Research*, pp. 1209–1230, 2014.

[39] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, “Nonlinear Optimal Control via Occupation Measures and LMI-Relaxations,” *SIAM Journal on Control and Optimization*, vol. 47, no. 4, pp. 1643–1666, 2008. [Online]. Available: https://doi.org/10.1137/070685051

[40] A. Vannelli and M. Vidyasagar, “Maximal lyapunov functions and domains of attraction for autonomous nonlinear systems,” *Automatica*, vol. 21, no. 1, pp. 69–80, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109885900998

[41] D. Grigoriev and N. Vorobjov, “Complexity of Null- and Positivstellensatz proofs,” *Annals of Pure and Applied Logic*, vol. 113, no. 1-3, pp. 153–160, 2001.

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

[43] C. N. D. Alexander Prestel, *Positive Polynomials*. Springer Berlin, Heidelberg, 2004.

[44] G. Valmorbida and J. Anderson, “Region of attraction analysis via invariant sets,” *Proceedings of the American Control Conference*, pp. 3591–3596, 2014.

[45] J. LaSalle, “Some extensions of liapunov’s second method,” *IRE Transactions on Circuit Theory*, vol. 7, no. 4, pp. 520–527, 1960.

[46] A. A. Ahmadi and P. A. Parrilo, “Stability of Polynomial Differential Equations: Complexity and Converse Lyapunov Questions,” 2013. [Online]. Available: http://arxiv.org/abs/1308.6833

[47] A. A. Ahmadi, S. Dash, and G. Hall, “Optimization over structured subsets of positive semidefinite matrices via column generation,” *Discrete Optimization*, vol. 24, pp. 129–151, 2017.

[48] P. Seiler and G. J. Balas, “Quasiconvex Sum-of-Squares Programming,” *IEEE Conference on Decision and Control*, no. 2, pp. 3337–3342, 2010.

[49] F. Hamidi, M. Aloui, H. Jerbi, M. Kchaou, R. Abbassi, D. Popescu, S. B. Aoun, and C. Dimon, “Chaotic particle swarm optimization for enlarging the domain of attraction of polynomial nonlinear systems,” *Electronics*, vol. 9, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/10/1704

[50] L. Khodadadi, B. Samadi, and H. Khaloozadeh, “Estimation of region of attraction for polynomial nonlinear systems: A numerical method,” *ISA Transactions*, vol. 53, no. 1, pp. 25–32, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.isatra.2013.08.005

[51] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2000.

[52] O. Hachicho and B. Tibken, “Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments,” *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, no. December, pp. 3150–3155, 2002.

[53] P. Polcz and G. Szederkényi, “Lyapunov function computation for autonomous systems with complex dynamic behavior,” *European Journal of Control*, vol. 65, 2022.

[54] P. Polcz, T. Péni, and G. Szederkényi, “Improved algorithm for computing the domain of attraction of rational nonlinear systems,” *European Journal of Control*, vol. 39, pp. 53–67, 2018.

[55] P. Polcz, T. Péni, B. Kulcsár, and G. Szederkényi, “Induced L2-gain computation for rational LPV systems using Finsler’s lemma and minimal generators,” *Systems and Control Letters*, vol. 142, 2020. [Online]. Available: https://doi.org/10.1016/j.sysconle.2020.104738

[56] P. Polcz, T. Péni, and G. Szederkényi, “Computational method for estimating the domain of attraction of discrete-time uncertain rational systems,” *European Journal of Control*, vol. 49, pp. 68–83, 2019.

[57] B. Reznick, “Some concrete aspects of Hilbert’s 17th problem, Contemporary Mathematics ,” *American Mathematical Society, Providence, RI*, vol. 253, pp. 251–272, 2000.

[58] M. Choi, T. Lam, and B. Reznick, *Sums of squares of real polynomials*, ser. Proceedings of Symposia in Pure Mathematics. United States: American Mathematical Society, 1995, pp. 103–126.

[59] J. G. Vanantwerp and R. D. Braatz, “Tutorial on linear and bilinear matrix inequalities,” *Journal of Process Control*, vol. 10, no. 4, pp. 363–385, 2000.

[60] O. Toker and H. Ozbay, “On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback,” *Proceedings of the American Control Conference*, vol. 4, pp. 2525–2526, 1995.

[61] V. Blondel and J. N. Tsitsiklis, “NP-hardness of some linear control design problems,” *SIAM Journal on Control and Optimization*, vol. 35, no. 6, pp. 2118–2127, 1997.

[62] M. Kojima, “Sums of squares relaxations of polynomial semidefinite programs,” Tokyo Inst. of Technology, Tech. Rep., 2003. [Online]. Available: http://www.optimization-online.org/DB_FILE/2003/11/772.pdf

[63] P. A. Parrilo, “Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization,” *PhD thesis, California Institute of Technology, Pasadena, CA*, vol. 2000, p. 117, 2000. [Online]. Available: https://resolver.caltech.edu/CaltechETD:etd-05062004-055516

[64] D. Henrion, J. B. Lasserre, and C. Savorgnan, “Approximate volume and integration for basic semialgebraic sets,” *SIAM Review*, vol. 51, no. 4, pp. 722–743, 2009. [Online]. Available: https://doi.org/10.1137/080730287

[65] A. Gasull, H. Giacomini, and J. Torregrosa, “Explicit non-algebraic limit cycles for polynomial systems,” *Journal of Computational and Applied Mathematics*, vol. 200, no. 1, pp. 448–457, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042706000276

[66] J. F. Sturm, “Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999. [Online]. Available: https://doi.org/10.1080/10556789908805766

[67] P. Seiler, “SOSOPT: A Toolbox for Polynomial Optimization,” *ArXiv*, 2013. [Online]. Available: http://arxiv.org/abs/1308.1889

[68] M. Jones and M. M. Peet, “Using SOS for optimal semialgebraic representation of sets: Finding minimal representations of limit cycles, chaotic attractors and unions,” *Proceedings of the American Control Conference*, pp. 2084–2091, 2019.

[69] M. Tsatsos, “The van der pol equation,” *ArXiv*, 2008. [Online]. Available: https://arxiv.org/abs/0803.1658

[70] D. Li, D. Ignatyev, A. Tsourdos, and Z. Wang, “Estimation of non-symmetric and unbounded region of attraction using shifted shape function and r-composition,” *ArXiv*, 2022. [Online]. Available: http://arxiv.org/abs/2207.05421

[71] A. A. Ahmadi, M. Krstic, and P. A. Parrilo, “A Globally Asymptotically Stable Polynomial Vector Field with no Polynomial Lyapunov Function,” *IEEE Conference on Decision and Control and European Control Conference*, no. 2, pp. 7579–7580, 2011.

[72] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali, *LMI control toolbox user’s guide*. The Mathworks Partner Series, 1995.

[73] B. A. Coomes, “The Lorenz system does not have a polynomial flow,” *Journal of Differential Equations*, vol. 82, no. 2, pp. 386–407, 1989.

[74] S. S. Hassan, M. P. Reddy, and R. K. Rout, “Dynamics of the Modified n-Degree Lorenz System,” *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 2, pp. 315–330, 2019.

[75] D. Jones, “Stability analysis of the chaotic lorenz system with a state-feedback controller,” *Electronic Theses and Dissertations*, 2009. [Online]. Available: https://digitalcommons.georgiasouthern.edu/etd/944

[76] J. Konigsmarkova and M. Schlegel, “Identification of n-link inverted pendulum on a cart,” *Proceedings of the International Conference on Process Control*, pp. 42–47, 2017.

[77] S. Lam and E. J. Davison, “The real stabilizability radius of the multi-link inverted pendulum,” *Proceedings of the American Control Conference*, pp. 1814–1819, 2006.

[78] J. P. Parker, D. Goluskin, and G. M. Vasil, “A study of the double pendulum using polynomial optimization,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 10, 2021. [Online]. Available: https://doi.org/10.1063/5.0061316

[79] Y. Ilyashenko, “Centennial history of Hilbert’s 16th problem,” *Bulletin of the American Mathematical Society*, vol. 39, no. 3, pp. 301–354, 2002.

[80] A. Gasull and H. Giacomini, “Effectiveness of the Bendixson-Dulac theorem,” *Journal of Differential Equations*, vol. 305, pp. 347–367, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022039621006239

[81] J. S. Muldowney, “Compound matrices and ordinary differential equations,” *Rocky Mountain Journal of Mathematics*, vol. 20, no. 4, pp. 857–872, 1990.

[82] R. Ofir and M. Margaliot, “The Multiplicative Compound of a Matrix Pencil with Applications to Difference-Algebraic Equations,” 2021. [Online]. Available: http://arxiv.org/abs/2111.01419

[83] O. I. Kleshchina, “Some properties of the Lozinskii logarithmic norm,” *Differential Equations*, vol. 53, no. 6, pp. 709–718, 2017.

[84] R. A. Smith, “Some Applications of Hausdorff dimension inequalities for ordinary differential equations,” *Proc. Royal Society of Edinburgh: Section A Mathematics*, vol. 104, no. 3-4, pp. 235–259, 1986.

[85] C. W. Scherer and C. W. Hol, “Matrix sum-of-squares relaxations for robust semi-definite programs,” *Mathematical Programming*, vol. 107, no. 1-2, pp. 189–211, 2006.

APPENDIX A

Proofs

Theorem A.1. *Consider the autonomous dynamical system characterized by*

$$\dot{\mathbf{x}} = -\mathbf{x} + (\mathbf{x}^\top \mathbf{B}\mathbf{x})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{B} \succ 0 \tag{A.1}$$

The region of attraction of system (A.1) associated with the equilibrium point $\mathbf{x} = 0$ *is given by* $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^\top \mathbf{B}\mathbf{x} < 1\}$ *and also a possible Lyapunov function for the system is* $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{B}\mathbf{x}$*.*

Proof. It can be easily seen that the equilibrium points of system (A.1) are $\mathbf{x} = 0$ and the points satisfying $\mathbf{x}^\top \mathbf{B}\mathbf{x} = 1$, meaning that no point outside the closed-set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^\top \mathbf{B}\mathbf{x} < 1\}$ belongs to the region of attraction. Let's assume the Lyapunov function of system (A.1) as $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{B}\mathbf{x}$ which is by definition positive everywhere except for $\mathbf{x} = 0$. The time derivative of V is as follows,

$$\begin{aligned} \dot{V} &= (\nabla_{\mathbf{x}} V) \cdot \dot{\mathbf{x}} = \nabla_{\mathbf{x}} (\mathbf{x}^\top \mathbf{B}\mathbf{x})(-\mathbf{x} + (\mathbf{x}^\top \mathbf{B}\mathbf{x})\mathbf{x}) \\ &= 2\mathbf{x}^\top \mathbf{B}(-\mathbf{x} + (\mathbf{x}^\top \mathbf{B}\mathbf{x})\mathbf{x}) \\ &= 2\mathbf{x}^\top \mathbf{B}\mathbf{x}(\mathbf{x}^\top \mathbf{B}\mathbf{x} - 1) \end{aligned} \tag{A.2}$$

The time derivative of V is negative for all points belonging to $\{\mathbf{x} \in \mathbb{R}^n \setminus \{0\} \mid \mathbf{x}^\top \mathbf{B}\mathbf{x} < 1\}$ and consequently $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^\top \mathbf{B}\mathbf{x} < 1\}$ is the region of attraction of (A.1). □

Theorem A.2. *Consider the autonomous dynamical system characterized by*

$$\dot{\mathbf{x}} = -\mathbf{x} + (\mathbf{z}^\top \mathbf{B}\mathbf{z})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{B} \succ 0 \tag{A.3}$$

where $\mathbf{z}$ *corresponds to a vector of monomials of* $\mathbf{x}$ *from degree 1 until m. The region of attraction associated with the equilibrium point* $\mathbf{x}_e = 0$ *is* $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$ *if*

$$\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}} \mathbf{z})\mathbf{x} > 0 \tag{A.4}$$

holds for all $\mathbf{x} \in \{\mathbf{x} \in \mathbb{R}^n \setminus \{0\} \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$.

Proof. Let's assume the Lyapunov function $V(\mathbf{x}) = \mathbf{z}^\top \mathbf{B}\mathbf{z}$. The time derivative of V is as follows,

$$\begin{aligned}\dot{V} &= (\nabla_{\mathbf{x}} V) \cdot \dot{\mathbf{x}} \\ &= \nabla_{\mathbf{z}}(\mathbf{z}^\top \mathbf{B}\mathbf{z})(\nabla_{\mathbf{x}}\mathbf{z}) \cdot \dot{\mathbf{x}} \\ &= 2\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z}) \cdot \dot{\mathbf{x}} \\ &= 2\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x}(\mathbf{z}^\top \mathbf{B}\mathbf{z} - 1)\end{aligned} \tag{A.5}$$

If all points satisfying $\mathbf{z}^\top \mathbf{B}\mathbf{z} < 1$, also satisfies $\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} > 0$, then $\dot{V}$ is negative definite in $\{\mathbf{x} \in \mathbb{R}^n \setminus \{0\} \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$ and consequently the set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$ is the region of attraction of system (A.3).

□

Lemma 7. *Consider the autonomous system characterized by*

$$\dot{\mathbf{x}} = -\mathbf{x} + (\mathbf{z}^\top \mathbf{B}\mathbf{z})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{B} \succ 0 \tag{A.6}$$

If $\mathbf{B}$ is a diagonal matrix, the region of attraction associated with the equilibrium point $\mathbf{x} = 0$ is given by $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$.

Proof. With theorem A.2, it is only necessary to show that $\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} > 0$ holds for all points satisfying $\mathbf{z}^\top \mathbf{B}\mathbf{z} < 1$.

$$\begin{aligned}\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} &= \begin{bmatrix} z_1 & \dots & z_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} (\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} \\ &= \begin{bmatrix} \lambda_1 z_1 & \dots & \lambda_m z_m \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{x}} z_1 \\ \vdots \\ \nabla_{\mathbf{x}} z_m \end{bmatrix} \mathbf{x} = \begin{bmatrix} \lambda_1 z_1 & \dots & \lambda_m z_m \end{bmatrix} \begin{bmatrix} \langle \nabla_{\mathbf{x}} z_1,\ \mathbf{x} \rangle \\ \vdots \\ \langle \nabla_{\mathbf{x}} z_m,\ \mathbf{x} \rangle \end{bmatrix} \\ &= \lambda_1 z_1 \langle \nabla_{\mathbf{x}} z_1,\ \mathbf{x} \rangle + \ldots + \lambda_m z_m \langle \nabla_{\mathbf{x}} z_m,\ \mathbf{x} \rangle \\ &= \lambda_1 k_1 z_1^2 + \ldots + \lambda_m k_m z_m^2 \ > \ 0, \quad \text{it can be easily shown that } k_i > 0,\ i = 1 : m\end{aligned}$$

□

Lemma 8. *Consider the autonomous system*

$$\dot{\mathbf{x}} = -\mathbf{x} + (\mathbf{z}^\top \mathbf{B}\mathbf{z})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{B} \succ 0 \tag{A.7}$$

where $\mathbf{z}$ is a vector of monomials of $\mathbf{x}$ of degree $m > 0$, the region of attraction associated with the equilibrium point $\mathbf{x} = 0$ is the set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$.

Proof. With theorem A.2, it is only necessary to show that $\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} > 0$ holds for all $\mathbf{x} \in \mathbb{R}^n$ satisfying $\mathbf{z}^\top \mathbf{B}\mathbf{z} < 1$. If $\mathbf{z}$ corresponds to monomials of $\mathbf{x}$ of degree $m > 0$, then

$$(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} = m\mathbf{z}$$

And by definition, m is positive so $m \cdot (\mathbf{z}^\top \mathbf{B}\mathbf{z})$ is positive definite. □

Lemma 9. *Consider the autonomous system*

$$\dot{\mathbf{x}} = -\mathbf{x} + (\mathbf{z}^\top \mathbf{B}\mathbf{z})\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{B} \succ 0 \tag{A.8}$$

where $\mathbf{z}$ is a vector of monomials of $\mathbf{x}$ of degrees $\{m_1, \ldots, m_p\}$. Let $l(m_i)$ be the number of monomials of degree m_i in $\mathbf{z}$. If $\mathbf{B}$ is a matrix of the following form,

$$\mathbf{B} = \operatorname{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_p), \quad \operatorname{size}(\mathbf{A}_i) = l(m_i) \times l(m_i)$$

then the region of attraction associated with the equilibrium point $\mathbf{x} = 0$ is the set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{z}^\top \mathbf{B}\mathbf{z} < 1\}$.

Proof. With theorem A.2, it is only necessary to show that $\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} > 0$ holds for all $\mathbf{x} \in \mathbb{R}^n$ satisfying $\mathbf{z}^\top \mathbf{B}\mathbf{z} < 1$. Let's consider without loss of generality that $\mathbf{z}$ can be written in the following form,

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{bmatrix}, \quad \mathbf{z}_i \text{ is the vector containing monomials of degree } m_i$$

If matrix $\mathbf{B}$ is a diagonal block matrix of the following form,

$$\mathbf{B} = \operatorname{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_p), \quad \operatorname{size}(\mathbf{A}_i) = l(m_i) \times l(m_i)$$

then

$$\mathbf{z}^\top \mathbf{B}(\nabla_{\mathbf{x}}\mathbf{z})\mathbf{x} = \mathbf{z}^\top \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_p \end{bmatrix} \begin{bmatrix} m_1\mathbf{I}_1 & & \\ & \ddots & \\ & & m_p\mathbf{I}_p \end{bmatrix} \mathbf{z} =$$

$$= \mathbf{z}^\top \begin{bmatrix} m_1\mathbf{A}_1 & & \\ & \ddots & \\ & & m_p\mathbf{A}_m \end{bmatrix} \mathbf{z} = \mathbf{z}^\top \mathbf{B}^* \mathbf{z} > 0$$

□

APPENDIX B

2-link Pendulum System

The inverted double link pendulum system with a single input that we consider is presented in Figure B.1, which corresponds to a simple 2-link pendulum on a moving car.

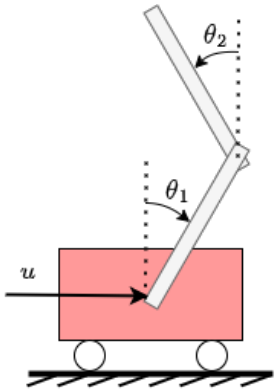


Figure B.1: Model of a 2-link inverted pendulum on a moving car.

The inverted 2-link pendulum on a car is a nonlinear system characterized by the second order differential equation

$$M(\mathbf{q})\ddot{\mathbf{q}} - F(\mathbf{q}, \dot{\mathbf{q}}) = u \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top \tag{B.1}$$

with $q = [d \quad \theta_1 \quad \theta_2]^\top$, being d the position of the car, θ_1 the angle of the bottom link with the vertical and θ_2 the angle made by the second link and the vertical.

We seek to rewrite equation (B.1) in order to obtain a first order system with polynomial representation, so that we can use the iterative sum-of-squares algorithms. For the case here in consideration, some values were chosen for the model such as mass and coefficient of fluid friction in each joint and the friction of the car.

Let us consider the vector state $\mathbf{x} = \mathbb{R}^4$ with $\mathbf{x} = [\theta_1 \quad \dot{\theta_1} \quad \theta_2 \quad \dot{\theta_2}]^\top$, and we obtain the following expression for $\mathbf{M}$ and $\mathbf{F}$.

$$M = \begin{bmatrix} 3 & 2\cos(x_1) & \cos(x_3) \\ 2\cos(x_1) & 3 & \cos(x_1 - x_3) \\ \cos(x_3) & \cos(x_1 - x_3) & 2 \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 2\sin(x_1)x_2^2 + \sin(x_3)x_4^2 \\ -\sin(x_1 - x_3)x_4^2 + 2\sin(x_1)g \\ \sin(x_1 - x_3)x_2^2 + \sin(x_3)g \end{bmatrix} \tag{B.2}$$

with $g = 10$.

Replacing matrices $\mathbf{M}$ and $\mathbf{F}$ into equation (B.1) and set on the left side the vector $\dot{\mathbf{x}}$, followed by a Taylor expansion of order four on the equilibrium point $\mathbf{x} = 0$ of the right side we get

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \left[f_1(\mathbf{x}, u)\ f_2(\mathbf{x}, u)\ f_3(\mathbf{x}, u)\ f_4(\mathbf{x}, u)\right]^\top, \quad \mathbf{x} \in \mathbb{R}^4 \quad \text{and} \quad u \in \mathbb{R} \tag{B.3}$$

with

$$\begin{aligned} f_1 =\ & x_2 \\ f_2 =\ & -0.975x_1^2x_2 + 0.125x_2^2x_3 - 0.625x_4^2x_1 + 0.25x_4^2x_3 - 13.02x_1^3 + 4.843x_1^2x_3 + 0.515ux_1^2 \\ & - 4.687x_1x_3^2 + 0.031ux_1x_2 + 12.5x_1 + 1.146x_3^3 - 0.031ux_3^2 - 1.25x_3 - 0.375u \\ f_3 =\ & x_4 \\ f_4 =\ & 0.375x_2^2x_1 - 0.625x_2^2x_3 + 0.125x_4^2x_1 - 0.25x_4^2x_3 + 3.854x_1^3 - 8.594x_1^2x_3 - 0,140ux_1^2 \\ & + 3.437x_1x_3^2 + 0.218ux_1x_3 - 2.5x_1 - 2.604x_3^3 + 0.093ux_3^2 + 6.25x_3 - 0.125u \end{aligned}$$

and with the linear feedback

$$u = -\begin{bmatrix} -297.273 & -70.775 & 300.439 & 115.000 \end{bmatrix} \mathbf{x}$$