Analysis of Nonlinear Systems: Large-Scale, Time-Varying, Non-Polynomial and Uncertain Properties

Edward Hancock

D.Phil Thesis



University of Oxford

Christ Church

Department of Engineering Science

Trinity 2013

Contents

1 Introduction **8**

1.1 Outline and Contribution 14

1.2 Examples and Applications 15

2 Background: Nonlinear Systems **18**

2.1 Introduction 18

2.2 Ordinary Differential Equations 19

2.3 Lyapunov Theory: Autonomous Systems 21

2.3.1 Stability Definitions 21

2.3.2 Lyapunov functions 22

2.3.3 Linear Systems 25

2.4 Lyapunov Theory: Time-varying Systems 26

2.4.1 Stability Definitions 26

2.4.2 Lyapunov functions 28

2.4.3 Linear Time-Varying Systems 28

2.5 Linear Matrix Inequalities 29

2.6 Absolute Stability 32

2.7 Sum of Squares 34

2.7.1 Lyapunov theory and SOS Programming 37

2.8 Conclusion 37

3 Generalised Absolute Stability and Sum of Squares **38**

3.1 Introduction 38

3.2 Functional Sum of Squares and Generalised Sectors 41

3.2.1 Functional Sum of Squares 41

3.2.2 Generalised Sector Conditions 44

3.2.3 Generalised Sectors for Integral Inequalities 46

3.3 Generalised Absolute Stability 47

3.3.1 Generalised Absolute Stability Definitions 47

3.3.2 Polynomial Lyapunov functions 51

3.3.3 Lur'e type Lyapunov functions . . . 52
3.4 Region of Attraction . . . 55
3.4.1 ROA using Polynomial Lyapunov function . . . 56
3.4.2 ROA using Lur'e type Lyapunov functions . . . 57
3.5 Further Examples . . . 59
3.6 Conclusion . . . 60

4 Perturbation Methods for Invariant Sum of Squares and Semidefinite Programming 62
4.1 Introduction . . . 62
4.2 Background: Invariant Polynomials and SDPs . . . 64
4.3 A Networked Symmetric Polynomial using a Representation of S_3 . . . 73
4.4 Perturbation Methods for Invariant SDP and Polynomials . . . 77
4.4.1 Generating Invariant SDPs and Polynomials: Necessary Conditions and Lower Bounds . . . 78
4.4.2 Unstructured Perturbations of Invariant Polynomials and SDPs: Sufficient Conditions . . . 80
4.4.3 Structured Perturbations of Invariant Polynomials and SDPs: Sufficient Conditions and Upper Bounds . . . 82
4.4.4 Block Diagonalisation . . . 84
4.4.5 Algorithms . . . 87
4.5 Networked Polynomials using S_N . . . 88
4.6 Examples . . . 93
4.6.1 Non negativity of Polynomials . . . 93
4.6.2 Lyapunov functions with a Network Structure . . . 95
4.7 Future Directions . . . 99
4.8 Conclusion . . . 101

5 Invariance Principles for Time-varying Systems 102
5.1 Introduction . . . 102
5.2 Background: Dynamical Systems and Invariance . . . 104
5.2.1 Dynamical Systems . . . 105
5.2.2 Continuous Dependence on Initial Conditions: Time-varying systems 105
5.2.3 The Invariance Principle . . . 106
5.2.4 Invariance-like Theorems for Time-varying Systems . . . 106
5.3 The Invariance Principle for Autonomous Systems . . . 108
5.4 Trajectories leaving Sets Uniformly . . . 111
5.4.1 Definitions and Properties of Trajectories leaving a Set . . . 111
5.4.2 Coordinate Transformations for Trajectories Leaving Sets . . . 114

5.4.3 Lyapunov Functions and Attractivity 117
5.5 Invariance Principles for Time-Varying Systems 118
5.5.1 Lyapunov Functions and Invariance 118
5.5.2 Non-Recursive Methods 120
5.5.3 Output Recursive Methods and Observability 122
5.5.4 Comparison to Existing Methods 123
5.6 Applications to Nonlinear, Large-Scale, Time-Varying Systems 124
5.7 Conclusion 125

6 Conclusion **127**

Acknowledgements

I would like to grateful acknowledge the support, guidance and advice of my supervisor, Dr Antonis Papachristoudoulou, without whom this thesis would not have been possible. I am also indebted to the Clarendon Fund, the Overseas Research Scholarships Award Scheme, and more recently the Bracken fund, which have enabled me to study at Oxford.

I would like to thank Prof Richard Murray and Prof John Doyle for enabling me to spend time as a visiting student at the California Institute of Technology during my studies. I would also like to thank Dr Malcolm McCulloch for giving me the chance to tutor during my DPhil.

I would like to thank Johannes Buerger, James Anderson and the DPhil students and Postdocs in the control group for their help and company during much needed coffee breaks and pub visits. I would also like to thank Oliver Murphy, Chris Noon, Stefan Sienkiewicz, Kerrith Davies, Claire and Anass El Mouden and the rest of the Christ Church GCR for making the past few years such enjoyable ones.

I am indebted to Suzanne English for all of her love and support during my DPhil. I would like to thank my sisters Amy and Olga Hancock, for their support. Finally, I would like to dedicate this thesis to my parents, Sue and Greg Hancock, for their love and encouragement during all of my studies.

Notation

$\mathbb{R}^+$: the set of non-negative numbers

$[a, b]$: the closed interval between real numbers a and b

$\mathbb{R}^n$: the set of vectors with n elements

$\mathbb{R}^{m\times m}$: the set of $m \times m$ matrices

∂M: the boundary of set M

$\overline{M}$: the closure of set M

$f : D \to P$: function f mapping inputs from set D to outputs in set P

$f(x) \in \mathbb{R}^n[x]$: Shorthand for polynomial function f mapping x to n-dimensional output

$x \to b$: Shorthand for $x(t) \to b$ as $t \to \infty$.

$d(x, P) = \inf_{x_p \in P} \|x - x_p\|$.

$d_x((x, t), P) = \inf_{(x_p,t_p)\in P} \|x - x_p\|$.

$N_\alpha(P) = \{x \in D | d(x, P) \leq \alpha\}$.

$N^x_\alpha(P) = \{(x, t) \in D | d_x((x, t), P) \leq \alpha\}$.

A^T: the transpose of A

$A \succeq 0$ implies matrix A is a positive semidefinite matrix

$A \succ 0$ implies A is a positive definite matrix

$\langle X, Y\rangle = Tr(XY^T)$: Inner product or trace of matrices X and transpose of Y.

$\|\cdot\|$: the Euclidean norm

$\Re\{x\}$ the real part of complex number x

$\mathbf{S}^N$: the vector space of real symmetric $N \times N$ matrices

$\mathbf{S}^N_+$: the positive semi-definite subset of $\mathbf{S}^N$

$GL(V)$: the general linear group over vector space V

$\sigma(g), \sigma_g$: A linear representation $\sigma : g \to GL(V)$ for vector space V

$|G|$: the order of the group G

Abstract

Analysis of Nonlinear Systems: Large-Scale, Time-Varying, Non-Polynomial and Uncertain Properties

Edward Hancock

D.Phil. Thesis

Christ Church

Trinity 2013

This thesis introduces, develops and applies methods for analysing nonlinear systems with the multiple challenges of time-varying, non-polynomial, uncertain or large-scale properties. Both computational and analytic methods using Lyapunov functions are developed and the methods are applied to a range of examples. Generalised Absolute stability is introduced, which is a method of treating polynomial systems with non polynomial, uncertain or time-varying feedback. Analysis is completed with Sum of Squares programming, and this method extends both the applicability of sum of squares as well as existing absolute stability theory. Perturbation methods for invariant Sum of Squares and Semidefinite programs are introduced, which significantly improves scalability of computations and allows sum of squares programming to be used for large scale systems. Finally, invariance principles are introduced for nonlinear, time-varying systems. The concept of trajectories leaving sets uniformly in time is introduced, which allows a non-autonomous version of Barbashin-Krasovskii theorem.

Chapter 1

Introduction

Nonlinear systems theory is essential for understanding a diverse range of subjects in science, technology and economics [41, 62, 76, 79]. It is required to understand how systems function and fail in everything from critical infrastructure, such as power grids [17, 58] and the internet [65], to population dynamics [62] and chemical reaction networks [42]. Systems such as these need to involve models with relationships which are inherently not linear. For example, in understanding how power grids can lose stability and fail, nonlinear relationships modelling electrical generators and power lines need to be used [17, 58]. However, although nonlinear systems have been extensively studied [41, 79], many challenging problems remain in analysing and understanding these systems, particularly when nonlinearity is combined with other difficult-to-analyse properties, such as large-scale systems.

Although linear systems theory can be informative for nonlinear systems analysis, nonlinear systems can exhibit phenomena which cannot exist in linear systems [41]. Linear systems are simpler and so preferable to analyse over nonlinear systems as they can be decomposed into parts, which are solved independently and then recombined, whereas this property does not hold for nonlinear systems [79]. This simplification can be informative for analysis of nonlinear systems, which can be approximated by simpler linear systems for a small range of the variables [41]. However, nonlinear systems can exhibit phenomena

such as multiple isolated equilibrium points, limit cycles, or even chaos [41, 79]. Therefore, nonlinear systems need to be taken into account to understand what behaviour can occur, when each behaviour occurs and how the system functions for a large range of variables. As a simple example in molecular biology, it is possible to have multiple equilibrium points in a genetic system, modelled by concentrations of proteins produced from particular genes [5]. This gives cells with the same genome the ability to function in a number of different ways, even in the same organism, e.g. cell differentiation of skin, muscle and liver cells in the body [5]. By using linear models, it is possible to understand the behaviour of this system near each equilibrium point, but a nonlinear model is required to understand when each type of behaviour occurs and how the overall system functions [41, 79].

There are many system properties beyond nonlinearity that can make analysis of systems both conceptually and computationally challenging. In this thesis, nonlinear systems composed of Ordinary Differential Equations (ODEs) are studied, where a finite number of variables change continuously with time and the differential equations describe the rate of change of the variables [29]. In particular, this thesis studies the case for nonlinear ordinary differential equations where the ODEs are only known to some certainty (uncertain), where the ODEs change over time (time-varying), where the ODEs have non-polynomial terms (non-polynomial) or where the system has a large number of variables (large-scale) [29]. For example, an aircraft may be modelled by a finite number of variables and nonlinear ODEs to describe its orientation and velocity [6]. The aircraft model may be uncertain due uncertain parameters and unmodelled dynamics [6] and time-varying due to changes in configuration or aerodynamic coefficients at different 'operating points' in the 'flight envelope' [6]. Further, with a number of (automated) aircraft flying in formation the model may be a large-scale networked system [4]. Time-varying systems also arise from the mathematical 'trick' of transforming variables of an autonomous (time-invariant) system in order to simplify analysis of system behaviour not at an equilibrium point, e.g. periodic oscillations or chaos. The added time-varying, uncertain or large-scale properties make a

nonlinear system much more difficult to analyse, both due to the conceptual difficulty as well as the difficulty in computing or carrying out the necessary analysis methods. For a large-scale networked system, such as the Internet or a power system, it might be both difficult to conceptually relate its structure to its behaviour [19, 88], or it may be difficult for methods to be used beyond a certain number of variables due to scaling issues, no matter what the structure of the network [84].

A key property required to understand how systems function and fail is to be able to understand and analyse system stability. When studying nonlinear systems, we typically cannot solve the equations, which are analytically intractable [79]. However, we only need to know qualitative and quantitative properties of a system rather than the explicit solution to understand its behaviour, such as whether the system settles down to an equilibrium after a long period of time [79]. A first step in understanding a nonlinear system is to determine equilibrium points and any other possibilities, such as periodic oscillations [41]. The second step is to determine whether an equilibrium is stable, by which we mean whether the system states approach the equilibrium after a long period of time and whether states starting sufficiently close to one of these equilibriums will stay close [41]. For example, a damped, oscillating pendulum is stable if for small initial oscillations it eventually stops oscillating and it does not obtain large oscillations during the transient period. It can be noted that stability of other types of behaviours such as periodic oscillations can be framed in terms of stability of equilibrium [41]. A system that has two or more stable equilibrium points is a case where it is useful to determine for which conditions the system will finish at each equilibrium [41]. Finding which equilibrium the system will be attracted to involves estimating regions of attraction, which determine how close the initial conditions have to be to the equilibrium or other behaviour in order for the system to approach that equilibrium after a long period of time. For the above mentioned molecular biology example regarding cell differentiation, the genetic system must have two or more stable equilibrium points, each representing a different cell type. For this example, the stability allows the skin, muscle or liver cells and their offspring to

'remember' their cell type [62], as a system near one equilibrium will not switch to the other. This is an example, where it is useful to find for which initial conditions of protein concentrations the different types of cells will occur. Stability is also a necessary stepping stone to analysing other important properties such as finding whether the system is robust to noise and uncertainty, or how fast the system converges to the equilibrium [29, 41].

In this thesis, Lyapunov theory is used to analyse system stability, which has a number of advantages over other analysis methods, particularly for nonlinear systems which are time-varying, uncertain or large scale. Lyapunov theory uses 'generalised energy' to analyse the dynamic behaviour of a nonlinear system, allowing it to be analysed in a general, robust and informative way. As a simple example of 'generalised energy', we may study the dynamics of an oscillating pendulum, such that if the total physical energy of a pendulum decreases over time then we know that the pendulum will eventually stop. Similarly, we can use a 'generalised' energy for different systems such as population biology [37]. If a Lyapunov function can be found, then this gives sufficient stability conditions for the system, where we know for which parameter values a system (such as the pendulum) is stable, but cannot make a statement on stability for parameters outside that range. Lyapunov theory can be used to analyse both local and non-local behaviour about an equilibrium point [41]. This is in contrast to other techniques, such as eigenvalue methods for the linearisation, which can determine whether a particular equilibrium point in a nonlinear system is stable or unstable but cannot determine the behaviour further away from the equilibrium point. Furthermore, Lyapunov theory can be used to study systems with other challenging properties, such as nonlinear time-varying systems, where eigenvalue analysis and other linear techniques cannot be used even for the linear case [41].

The key challenge in using Lyapunov theory is to find a Lyapunov function for a system, which can either be found analytically or computationally [41]. For electrical or mechanical systems, a natural choice of Lyapunov function is the physical energy. However, for other systems, finding a Lyapunov function analytically can be more challenging [41]. The

difficulty in finding Lyapunov functions often results in conservative stability conditions for a system. For this reason, computational methods are often preferred, as they can make finding a Lyapunov function far simpler and reduce conservatism. Similarly, it is often possible to find a Lyapunov function locally, but it can be more difficult to find a Lyapunov function providing a good region of attraction estimate. On the other hand, if a Lyapunov function can be found analytically, then it can often be more scalable for large systems and more informative than an equivalent computational approach.

A theoretical challenge to implementing Lyapunov theory analytically is to show stability for the cases where the Lyapunov function is not always decreasing with time, but instead is non-increasing, as is typically the case for the choice of mechanical energy of a system. For example, for the oscillating pendulum, the energy of the pendulum is decreasing while the pendulum is swinging due to damping, but is not decreasing for the instant when the pendulum stops and starts swinging in the other direction. This theoretical issue has been resolved by using Krasovskii-Lasalle's Invariance Principle for the autonomous (time-invariant) case [15, 45, 47, 48]. This makes finding a Lyapunov function analytically to show stability simpler and more practical. However, equivalent theory for the time-varying case which is relatively simple and practical to implement requires further development.

Historically, a number of computational methods were developed to extend Lyapunov methods for linear systems to nonlinear systems, but these often give conservative stability conditions for large variable ranges. Using systems theory, this is typically achieved by treating a nonlinear system as a linear system with feedback, where the feedback is used to represent all of the difficult to analyse terms. Absolute Stability theory studies stability of systems where the nonlinearity is in a sector bounded by two linear functions. This allows linear systems tools to be applied to the problem. Stability criteria are then found using two choices of Lyapunov function, resulting in the Circle and Popov criteria [41]. For simple systems this allows the stability criteria to be used geometrically [41]. Efficient computational methods were later developed using Linear Matrix Inequalities, a standard form of algebraic inequalities when evaluating Lyapunov functions for linear

systems [20]. These can be extended to nonlinear systems using the absolute stability framework [20, 34]. Similarly, under the framework of Integral Quadratic Constraints, efficient computational methods for Linear Matrix Inequalities have been developed to study nonlinear systems [60]. However, often these results give conservative stability conditions for larger ranges of the variables as they treat the nonlinear feedback as an uncertainty and so important information about the nonlinearity may be lost.

Sum of Squares programming is a more recent computational method for testing non-negativity of polynomials and has proved useful for finding Lyapunov functions for polynomial systems [25, 66, 68]. Testing whether a polynomial function is a sum of squares can be converted into a Linear Matrix Inequality, for which computational methods exist, as mentioned above. The method extends previous computational methods by finding Lyapunov functions for a large class of nonlinear systems directly, rather than by breaking the differential equations into a linear system with nonlinear feedback. It thus allows the analysis to give less conservative stability conditions for a larger range of the variables [25, 68]. This technique has been shown to be useful for stability, region of attraction and robustness analysis for nonlinear systems, particularly as domains extend further from the equilibrium point [25, 68].

To significantly expand on previous successes, further work is required on Sum of Squares programming in the three areas of new frameworks, scalability and non-convex versions. Firstly, new frameworks need to be developed for analysis with Sum of Squares programming, where these may have previously been impractical when using either analytic methods or existing computational tools. Secondly, the current major limitation of using Sum of Squares programming is that it can typically only be used for systems with 10-15 variables [84]. Thus work is required in order to apply Sum of Squares programming to large-scale systems. Thirdly, when using Sum of Squares for optimisation, such as finding the largest region of attraction estimate, it is possible for the underlying problem to be a bilinear matrix inequality, a form of algebraic inequality for which less efficient computational methods exist. However, the third problem is a wider problem in both systems

and control theory and optimisation, and not unique to sum of squares. In this thesis we focus on developing new frameworks and scaling Sum of Squares. This has potential applications in areas as diverse as systems biology and power systems analysis.

1.1 Outline and Contribution

In this thesis, Lyapunov theory is developed and implemented for nonlinear systems with time-varying, non-polynomial, uncertain and large-scale properties. These 'multiple challenging properties' can make analysis difficult when treated individually, but by using new conceptual approaches or by using computational tools in new ways, the state-of-the-art can be significantly extended.

The outline of the thesis is as follows:

- In Chapter 2, a background for nonlinear systems as well as sum of squares programming is presented, as these are used throughout the thesis. Chapter specific background on group representation theory is presented in Chapter 4 and invariance in dynamical systems is presented in Chapter 5.

- In Chapter 3, an extension of Absolute Stability theory is proposed, which uses the framework of a polynomial system with a feedback term containing non-polynomial, time-varying or otherwise difficult to analyse terms. The proposed framework is analysed using sum of squares programming. It extends existing absolute stability theory and also extends the applicability of sum of squares programming. This work was presented in [32].

- In Chapter 4, perturbation methods for symmetric or invariant problems are developed for Sum of Squares (SOS) and Semidefinite Programming (SDP). By perturbing invariant SOS and SDP problems, it is possible to extend the technique to non-invariant polynomials and SDPs. This allows analysis which significantly improves the scalability of SOS and SDP techniques, as well as providing other

improvements for numerical calculation. We apply the technique on a number of examples of polynomials and polynomial Lyapunov functions with a network structure in order to show the practical application. This work was partly presented in [30].

- In Chapter 5, new invariance principles for time-varying systems are proposed. The principles simplify multiple existing methodologies, which allows a practical application to more challenging problems as well as a clearer elucidation of unifying principles between the various techniques. The work also provides an alternative theoretical perspective to the existing literature. This work was partly presented in [31].

- In Chapter 6, a conclusion of the thesis is provided and future directions are discussed.

There are a number of common themes across the thesis. Firstly, all chapters aim to develop methods for finding Lyapunov functions, by either analytical or computational methods, as this is a primary challenge of applying Lyapunov theory. A second common theme is the analysis of nonlinear systems with 'multiple' challenging properties, such as nonlinear systems which are also large scale, non-polynomial, time-varying or uncertain. Finally, Sum of Squares programming, a computational method for nonlinear systems, is developed and applied in Chapters 3 and 4.

1.2 Examples and Applications

Throughout this thesis the examples of pendula, coupled oscillators or similar systems are regularly used to illustrate the practical applications and usefulness of the theory. Pendula provide a useful prototypical example as they are both pedagogically useful, being conceptually simple but displaying a range of behaviour, and they can also be used to represent a wide range of physical systems, including power grids, Josephson junctions,

electrical circuits and mechanical systems [13, 17].

A standard model for the pendulum is [41]

$$\ddot{\theta} + \gamma\dot{\theta} + g\sin\theta = 0 \tag{1.1}$$

where $\theta = \theta(t)$ is the angle of the pendulum, γ is a damping constant and g is a swing constant. The $\sin\theta$ term adds a nonlinearity to the system. A forcing term $p(t)$ can be added with

$$\ddot{\theta} + \gamma\dot{\theta} + g\sin\theta = p(t) \tag{1.2}$$

When p is a constant then (1.2) is closely related to the equations used for power system stability, and so is useful as a test case for new methods of analysing power systems. In particular, finding the set of initial conditions $(\theta(0), \dot{\theta}(0))$ for which the system is stable, or region of attraction analysis, is closely related to determining power system stability during faults [41].

This standard model of a pendulum motivates two questions in the thesis. Firstly, how can (1.2) be analysed using Sum of Squares programming, which requires the equations to be polynomial, whereas (1.2) contains a $\sin\theta$ term, which is non-polynomial. A proposed solution to this question is given in Chapter 3. Secondly, how can the stability of (1.1) be analysed using analytical methods if $\gamma = \gamma(t)$ and $g = g(t)$ are time-varying. A method to study this question is proposed in Chapter 5.

A pendulum-like oscillator with a polynomial nonlinearity, which is a modified Duffing oscillator [41], is described by

$$\ddot{\theta} + \gamma\dot{\theta} + g(\theta - \theta^3) = p(t) \tag{1.3}$$

In order to test methods for large scale systems analysis, we use coupled Duffing oscillators, where the coupling is in the structure of a network. This allows us to develop SOS methods to deal with large scale systems independently of developing methods for non-polynomial

systems. The coupling takes the place of the forcing term, where the system is represented by

$$\ddot{\theta}_i + \gamma_i \dot{\theta}_i + g_i(\theta_i - \theta_i^3) = \sum_j b_{ij}(\theta_j - \theta_i)^3 \tag{1.4}$$

for $i = 1, \ldots, N$, where N is the number of oscillators and b_{ij} represents a network. The coupling is chosen to be nonlinear for testing purposes, rather than for applications. In Chapter 4, equation (1.4) is used as a test case for stability methods using large-scale sum of squares algorithms.

Chapter 2

Background: Nonlinear Systems

2.1 Introduction

Understanding the properties of nonlinear systems, and in particular stability in nonlinear systems, is key to understanding how dynamical systems in biology, economics and technology function and fail. In this chapter, background is provided regarding nonlinear systems theory, focussing on stability of nonlinear systems. Ordinary differential equations (ODEs) are introduced before Lyapunov theory is presented, a method which uses 'generalised energy' to study stability of ODEs. Lyapunov theory is presented for both the autonomous and time-varying cases. Absolute Stability theory, Linear Matrix Inequalities and Sum of Squares programming are also introduced, which are frameworks combining theory and computational methods that allow Lyapunov theory to be implemented algorithmically.

Note: This Chapter provides background on nonlinear systems and sum of squares programming as these are required for multiple chapters. Background specific to individual chapters is provided in Chapter 4 on group representation theory and in Chapter 5 on invariance in dynamical systems. Proofs of theorems in this chapter are only provided if in the thesis the results/proofs are modified or new proofs are provided. For proofs of

Lyapunov Theory, see [41].

2.2 Ordinary Differential Equations

In this thesis, nonlinear systems composed of Ordinary Differential Equations (ODEs) are studied, where a finite number of variables change continuously with time and the differential equations describe the rate of change of the variables [29]. In particular, this thesis studies the case for nonlinear ordinary differential equations where the ODEs are only known to some certainty (uncertain), where the ODEs are non-polynomial, where the ODEs change over time (time-varying), or where the system has a large number of variables (large-scale) [29].

In this thesis, we consider Ordinary Differential Equation of the form

$$\dot{x} = f(x, t) \tag{2.1}$$

where $f : D \times \mathbb{R}^+ \to \mathbb{R}^n$ is locally Lipschitz in x uniformly in t, and piecewise continuous in t, D is an open, connected subset of $\mathbb{R}^n$, $x \in D$ and $x(0) = x_0$. Differential equations may vary with time (time-varying) or be independent of time (autonomous); equation (2.1) is a time-varying system as it is explicitly a function of t.

It is typically assumed that there is an equilibrium at the origin for (2.1), where $f(0, t) = 0$ for all t. This is a general assumption, as other cases may be transformed to this case. The transformation takes the form $z = x - \bar{x}$ with a reference trajectory $\bar{x}(t)$, such that $\dot{z} = \bar{f}(z, t) := f(\bar{x}(t) + z, t) - f(\bar{x}(t), t)$.

As an example of Ordinary Differential Equations in the form of (2.1), consider the 2nd order damped pendulum equation $\ddot{\theta} + \gamma\dot{\theta} + b\sin(\theta) = 0$, where θ is the angle of the pendulum, γ is the damping constant and b is the 'swing' constant. This equation can be

placed in the form of (2.1), where $x_2 = \theta$, $x_1 = \dot{\theta}$, and

$$\begin{aligned} \dot{x}_1 &= -\gamma x_1 - b\sin(x_2) \\ \dot{x}_2 &= x_1 \end{aligned} \tag{2.2}$$

The class of autonomous Ordinary Differential Equations is also considered, which takes the form

$$\dot{x} = f(x) \tag{2.3}$$

where $f : D \to \mathbb{R}^n$ is locally Lipschitz in x, D is an open, connected subset of $\mathbb{R}^n$, $x \in D$ and $x(0) = x_0$. Without loss of generality, it is typically assumed that $f(0) = 0$.

Time-varying differential equations, rather than autonomous equations, are extremely useful for modelling various systems. As mentioned in the introduction, an example of a time-varying system is an aircraft modelled by nonlinear ODEs to describe its orientation and velocity [6]. The aircraft model may be time-varying due to changes in configuration or aerodynamic coefficients at different 'operating points' in the 'flight envelope' [6]. Time-varying systems also often arise from the mathematical 'trick' of transforming variables of an autonomous (time-invariant) system in order to simplify analysis of system behaviour not at an equilibrium point, e.g. periodic oscillation or chaos. If $\bar{x}(t)$ is a solution of interest for (2.3), then the dynamics of nearby solutions can be studied by using the transformation $z = x - \bar{x}$, such that $\dot{z} = \bar{f}(z,t) := f(\bar{x}(t) + z) - f(\bar{x}(t))$, which is time-varying.

The continuity conditions on $f(x)$ and $f(x,t)$ are included to ensure that the system is 'well behaved' as long as x is bounded, as required by standard tools from dynamical systems theory. In this case, 'well behaved' solutions are those that exist, are unique, and that are continuously dependent upon initial conditions [41].

2.3 Lyapunov Theory: Autonomous Systems

When studying nonlinear systems, we typically cannot solve the differential equations analytically [79], however, we generally only need to know various qualitative and quantitative properties of a system to understand its behaviour [79]. In this section we first define these properties before introducing Lyapunov theory to analyse them. We consider the autonomous case, before looking at the time-varying case in the next section.

2.3.1 Stability Definitions

This thesis studies methods to analyse 'asymptotic stability', by which it is meant that the system trajectories approach the equilibrium after a long period of time and that trajectories starting sufficiently close to the equilibrium will stay close [41]. For example, a damped, oscillating pendulum is asymptotically stable if for small initial oscillations it eventually stops oscillating and it does not attain large oscillations during the transient period.

Attractivity is first defined in order to describe asymptotic behaviour.

Definition 2.1 *We say that the system trajectory $x(t) \to b$ as $t \to \infty$, if for all $\varepsilon > 0$ there exists $T \geq 0$ such that $\|x(t) - b\| \leq \varepsilon$ for all $t \geq T$.*

Definition 2.2 *The equilibrium point $x = 0$ of (2.3) is*

- *stable, if for all $\varepsilon > 0$, there exists $\delta = \delta(\varepsilon) > 0$ such that*

$$\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon, \ \forall t \geq 0 \tag{2.4}$$

- *unstable if it is not stable*
- *attractive if there exists a $\bar{\delta} > 0$ such that*

$$\|x(0)\| < \bar{\delta} \implies \lim_{t \to \infty} x(t) = 0 \tag{2.5}$$

- *asymptotically stable if it is attractive and stable*
- *globally asymptotically stable if it is stable and* $x \to 0$ *for all* $x(0) \in \mathbb{R}^n$

Asymptotic stability requires conditions for both the asymptotic behaviour through attractivity and the transient behaviour through stability.

As examples, for $x(t) = x_0 e^{-t}$ then $x = 0$ is asymptotically stable (attractive and stable), for $x(t) = x_0 \sin(t)$ then $x = 0$ is stable but not attractive, for $x(t) = x_0 e^{-t} + (e^{-t} - e^{-2t})$ then $x = 0$ is attractive but not stable, and for $x(t) = x_0 e^{t}$ then $x = 0$ is unstable. Similarly, for $x(t) = x_0 e^{-t}$ then $x = 0$ is globally asymptotically stable as $x \to 0$ for all $x(0)$.

2.3.2 Lyapunov functions

Lyapunov theory uses 'generalised energy' to analyse the dynamic behaviour of a nonlinear system in a general, robust and informative way. As a simple example of 'generalised energy', one may study the dynamics of an oscillating pendulum, where if the total physical energy of a pendulum decreases over time then the pendulum will eventually stop.

Lyapunov functions are 'bowl shaped' functions which decrease with time. Asymptotic stability is shown using Lyapunov functions as the system trajectories always end up in the bottom of the 'bowl'. Lyapunov functions provide a methodology to show that an equilibrium point is asymptotically stable.

We first define positive definite functions in order to test whether a function has a 'bowl' shape.

Definition 2.3 *Positive Definite function*

A function $V : D \to \mathbb{R}$ *is positive definite in* D *if* $V(0) = 0$ *and* $V(x) > 0$ *for all* $x \in D \backslash \{0\}$. V *is negative definite if* $-V$ *is positive definite. A function is positive semi-definite if* $V(0) = 0$ *and* $V(x) \geq 0$*, while a function is negative semi-definite if* $V(0) = 0$ *and* $V(x) \leq 0$*. A function is indefinite if it is not positive or negative semi-definite.*

As examples, $V(x) = x^2$ is positive definite in $\mathbb{R}$, $V(x, y) = x^2$ is positive semidefinite in $\mathbb{R}^2$, $V(x) = -x^2$ is negative-definite and $V(x) = 1 - x^2$ is indefinite.

Conditions for stability are given next by ensuring that the function is positive definite and decreasing with time.

Theorem 2.4 *Let $x = 0$ be an equilibrium point for (2.3) and let $D \in \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that*

$$V(x) \text{ is positive definite in } D \tag{2.6}$$

$$\dot{V}(x) \text{ is negative definite in } D \tag{2.7}$$

Then $V(x)$ is called a Lyapunov function and $x = 0$ in (2.3) is asymptotically stable.

Example 2.5 *For $\dot{x} = x(x - 1)$, if $V = \frac{1}{2}x^2$ then V is positive definite. Also, $\dot{V} = x^2(x - 1) < 0$ for $x < 1$ and so $x = 0$ is asymptotically stable by Theorem 2.4.*

A function $V(x)$ is called a Lyapunov function candidate if it has not yet been shown to be a Lyapunov function.

Showing asymptotic stability only guarantees the behaviour of trajectories close to the equilibrium and so extensions of Theorem 2.4 are required to understand the system dynamics further away from the equilibrium point.

Theorem 2.6 *Let $x = 0$ be an equilibrium point for (2.3) and let $D \in \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that*

$$V(x) \text{ is positive definite in } \mathbb{R}^n \tag{2.8}$$

$$V(x) \to \infty \text{ as } \|x\| \to \infty \tag{2.9}$$

$$\dot{V}(x) \text{ is negative definite in } \mathbb{R}^n \tag{2.10}$$

Then $V(x)$ is called a Lyapunov function and $x = 0$ in (2.3) is globally asymptotically

stable.

If a system is locally stable but not globally stable, then it can be useful to find the set of initial conditions for which $x \to 0$, or to find the 'region of attraction'.

To understand region of attraction analysis, we first note the definition of a positively invariant set.

Definition 2.7 *A set M is said to be positively invariant if $x_0 \in M$ implies $x(t) \in M$ for all $t \in \mathbb{R}_+$ and invariant if $x_0 \in M$ implies $x(t) \in M$ for all $t \in \mathbb{R}$.*

We next give the definition of a region of attraction.

Definition 2.8 *We define*

$$R_A = \{x_0 \in D | x(t) \to 0 \text{ as } t \to \infty\} \tag{2.11}$$

We define a region of attraction estimate to be a positively invariant subset of R_A.

Example 2.9 *For $\dot{x} = x(x-1)$ then it can be seen that $x \to 0$ for $x_0 < 1$, but this does not hold for all $x_0 \geq 0$, and so $R_A = \{x_0 \in \mathbb{R} | x_0 < 1\}$, which is positively invariant.*

If $x = 0$ is asymptotically stable and the region of attraction is $\mathbb{R}^n$, then $x = 0$ is globally asymptotically stable.

An estimate of the region of attraction can be found using Lyapunov functions, where the 'estimate' refers to a positively invariant set contained within R_A, such that all trajectories starting in the estimate eventually approach the origin.

Theorem 2.10 *[41] If the conditions for Theorem 2.4 hold and there exists $L > 0$ such that $L_V = \{x \in D | V(x) \leq L\}$ is bounded and $\dot{V} < 0$ for all $x \in L_V \backslash \{0\}$ then L_V is a subset of R_A.*

Example 2.11 *For $V = \frac{1}{2}x^2$ with the system in Example 2.9, then $\dot{V} < 0$ for $x < 1$ and so $x \to 0$ for all $\{x_0 \in \mathbb{R} | x_0^2 \leq L < 1\}$. This ROA estimate is a (strict) subset of the ROA (see Example 2.5).*

2.3.3 Linear Systems

A simpler approach to determine the local stability of the origin is to linearise the system about an equilibrium point. Linear systems are simpler to analyse as they can be decomposed into parts, which are solved independently and then recombined, whereas this does not hold for nonlinear systems [79]. In particular, eigenvalue analysis can be used on the linear system to determine the stability of an equilibrium point in the nonlinear system. This analysis is useful both in its own right and as a starting point for further analysis, such as for nonlinear systems.

Consider the linear system

$$\dot{x} = Ax \tag{2.12}$$

If (2.12) is a linearisation of a nonlinear system and the eigenvalues of A are all negative, then the equilibrium point of the nonlinear system is exponentially stable [41]. Lyapunov theory can obtain similar results for linear systems analysis.

Theorem 2.12 *Given a matrix $A \in \mathbb{R}^{m\times m}$, $Re\{\lambda_i\} < 0$ for all eigenvalues of matrix A if and only if for any given positive definite matrix Q, there exists a positive definite symmetric matrix P that satisfies*

$$PA + A^T P = -Q \tag{2.13}$$

Furthermore, P is unique.

If P exists then the origin of (2.3) is asymptotically stable.

Example 2.13 *For the pendulum in* (2.2) *where $\gamma, b \geq 0$, then*

$$P = \begin{pmatrix} 1 & \frac{1}{2}\gamma \\ \frac{1}{2}\gamma & b + \frac{1}{2}\gamma^2 \end{pmatrix}, \quad PA + A^T P = \begin{pmatrix} -\gamma & 0 \\ 0 & -\gamma b \end{pmatrix} \tag{2.14}$$

and so $x = 0$ in the system is asymptotically stable for $\gamma, b > 0$.

2.4 Lyapunov Theory: Time-varying Systems

Time-varying systems add a more challenging property for analysis, but Lyapunov theory can similarly be used for stability analysis for this case. The definitions and methodology are similar to the autonomous case, but an extra level of difficulty is required to deal with the time-varying nature of the system. For the definitions, this extra level is required to determine whether the stability conditions depend upon the starting time, while for the analysis, the Lyapunov function and its derivative may be time-varying.

2.4.1 Stability Definitions

For stability of a time-varying system, similar definitions are used to the autonomous case, with the added condition that stability and attractivity may be 'uniform'. This condition ensures that the stability and attractivity are independent of the initial time t_0. For stability, uniformity ensures that if a system starts close to the equilibrium point then it stays there, where the relationship between the 'starting' and 'staying' distances are independent of initial time. For attractivity, uniformity ensures that the rate of convergence and the initial conditions are independent of time.

Exponential stability is also introduced, which requires that the rate of convergence is exponential. This property can also hold for autonomous systems, but it is more important for time-varying systems when comparing stability of a nonlinear system with the linearisation.

Definition 2.14 *The equilibrium point $x = 0$ of (2.1) is*

- *stable, if for all $\varepsilon > 0$, there exists is $\delta = \delta(\varepsilon, t_0) > 0$ such that*

$$\|x(t_0)\| < \delta \implies \|x(t)\| < \varepsilon \tag{2.15}$$

for all $t \geq t_0$.

- *uniformly stable, if it is stable and $\delta = \delta(\varepsilon)$ is independent of t_0.*

- *unstable if it is not stable.*

- *attractive if there exists $\bar{\delta} = \bar{\delta}(t_0) > 0$ such that*

$$\|x(t_0)\| < \bar{\delta} \implies \lim_{t\to\infty} x(t) = 0 \tag{2.16}$$

- *uniformly attractive if for all $\bar{\varepsilon} > 0$, there exists $\bar{\delta}, T = T(\bar{\varepsilon}) > 0$ such that*

$$\|x(t_0)\| \leq \bar{\delta} \implies \|x(t)\| < \bar{\varepsilon} \tag{2.17}$$

  for all $t \geq t_0 + T$.

- *(uniformly) asymptotically stable if it is (uniformly) attractive and (uniformly) stable.*

- *exponentially stable if there exists $c, k, \bar{\delta} > 0$ such that*

$$\|x(t)\| \leq k\|x_0\|e^{-c(t-t_0)} \tag{2.18}$$

  for all $\|x_0\| < \delta$.

- *globally uniformly symptomatically stable if it is uniformly stable and $x \to 0$ uniformly in t for all $x_0 \in \mathbb{R}^n$.*

As examples, for $x(t) = x_0 e^{-(t-t_0)}$ then $x = 0$ is uniform asymptotically stable, whereas for $x(t) = \left[\frac{x_0^2}{1+2ax_0^2(t-t_0)}\right]^{1/2}$ from $\dot{x} = -ax^3$ then $x = 0$ is uniform asymptotically stable, but not exponentially stable. Similarly, if $a(t)$ is a slowly varying parameter for $\dot{x} = -(1-a(t)x^2)x$ such that $a(t) \to \infty$ then $x = 0$ is asymptotically stable, but not uniformly so, as the set of x_0 for which $x(t) \to 0$ becomes smaller for increasing t_0. The non-uniformity can be seen by noting that the region where $-(1 - a(t)x^2) < 0$, which is needed for asymptotic stability when $a(t)$ is slowly varying, becomes slowly smaller over

time. Thus the conditions for uniform attractivity do not hold.

2.4.2 Lyapunov functions

Lyapunov functions for time-varying systems have the shape of a 'valley' rather than a 'bowl' as in autonomous systems. In the time-varying case, x changes across the valley and t changes along the valley, while the equilibrium is represented by the bottom of the valley. Lyapunov functions are always decreasing with time and so the system trajectories are attracted to the bottom of the valley. The conditions ensure that the Lyapunov function does not 'flatten out', or become too steep to ensure uniformity.

Theorem 2.15 *[41, Theorem 4.9] Let $x = 0$ be an equilibrium point for (2.1) and D be a domain containing $x = 0$. Let $W_1(x), W_2(x), W_3(x)$ be continuous positive definite functions on D. Let $V : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ be a continuously differentiable function such that*

$$W_1(x) \leq V(x,t) \leq W_2(x) \tag{2.19}$$

$$\dot{V}(x,t) \leq -W_3(x) \tag{2.20}$$

for all $t \geq 0$ and all $x \in D$, then $x = 0$ is uniformly asymptotically stable. If $M \subset D$ is a compact set which is a neighbourhood of the origin and $0 < c < \min_{\partial M} W_1(x)$ then $x(t) \to 0$ uniformly in t for all $x_0 \in \{x \in D | W_2(x) \leq c\}$. If $D = \mathbb{R}^n$ and $W_1(x)$ is radially unbounded then $x = 0$ is globally uniformly asymptotically stable.

Example 2.16 *For $\dot{x} = -(1+\sin(t)^2)x$ then setting $V = \frac{1}{2}x^2$ gives $\dot{V} = -(1+\sin(t)^2)x^2 \leq -x^2 := -W_3(x)$, which shows that the system is uniformly stable.*

2.4.3 Linear Time-Varying Systems

Stability of time-varying systems can be analysed more simply by studying the linearisation of the system. These linearised systems can then be analysed with Lyapunov

functions.

Consider a system of the form

$$\dot{x} = A(t)x \tag{2.21}$$

where $f(x,t) = A(t)x$ in (2.1). If (2.21) is a linearisation about $x = 0$ and $x = 0$ is exponentially stable for the linear system, then it is also exponentially stable for the non-linear system [41]. We can study stability of linear time-varying systems using Lyapunov functions.

Theorem 2.17 *For $k_1, k_2, k_3 > 0$, let $P : \mathbb{R}^+ \to \mathbb{R}$ be a continuously differentiable function such that*

$$k_1 I \leq P(t) \leq k_2 I \tag{2.22}$$

$$P(t)A(t) + A(t)^T P(t) + \dot{P}(t) \leq -k_3 I \tag{2.23}$$

for all $t \geq 0$, then $x = 0$ in (2.21) is exponentially stable.

Example 2.18 *For the pendulum in (2.2) with bounded, time-varying $b = b(t)$, where $\gamma, b > 0$, then*

$$P = \begin{pmatrix} 1 & \frac{1}{2}\gamma \\ \frac{1}{2}\gamma & b + \frac{1}{2}\gamma^2 \end{pmatrix}, PA + \dot{P} + A^T P = \begin{pmatrix} -\gamma & 0 \\ 0 & -\gamma b + b'(t) \end{pmatrix} \tag{2.24}$$

and so the system is asymptotically stable for $-\gamma b + b'(t) \leq -\alpha < 0$.

2.5 Linear Matrix Inequalities

Linear Matrix Inequalities (LMIs) are a form of algebraic inequalities, which are useful for systems and control theory, and for which there exist efficient computational methods.

A Linear Matrix Inequality [20] has the form

$$F(y) = F_0 + \sum_{i=1}^{m} y_i F_i \succ 0 \tag{2.25}$$

where $y \in \mathbb{R}^m$ is the decision variable and the symmetric matrices $F_i \in \mathbb{R}^{n \times n}$ are given for $i = 0, \ldots, m$. Similarly, it is also possible to have a non-strict LMI, with the inequality $F(y) \succeq 0$.

LMIs are particularly useful for determining whether a quadratic function exists which is non-negative, as $f(x) = x^T Q x \geq 0$ if and only if $Q \succeq 0$.

An LMI is computationally feasible as $M = \{y \in \mathbb{R}^n | F(y) \succeq 0\}$ describes a convex set i.e. $F(\lambda y_1 + (1-\lambda) y_2) = \lambda F(y_1) + (1-\lambda) F(y_2) \succeq 0$ for all $\lambda \in [0, 1]$ and all $y_1, y_2 \in M$. This allows efficient computational methods to be employed based upon interior point algorithms for Semi-Definite Programming (SDP). Semi-definite programming optimises a cost function for a linear matrix constraint, where for given A, b, C the primal SDP is

$$\max_X Tr(C, X) \text{ such that } X \succeq 0 \text{ and } Tr(A_i, X) = b_i \text{ for } i = 1, \ldots, m \tag{2.26}$$

SDP also has the dual problem

$$\max b^T y \text{ such that } F - \sum_i y_i A_i \succeq 0 \tag{2.27}$$

SDPs can be solved efficiently with a purpose built software such as SDPT3 [73] or SEDUMI [80]. These programs often need parsers such as Yalmip [52] to convert the SDP from an easily written form to one useful for the SDP solver.

Example 2.19

$$Q = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} \succeq 0$$

for $\lambda = -3$. The search for λ for which $Q \succeq 0$ can be automated using efficient algorithms. When LMIs are used to determine Sum of Squares, it can be useful to obtain a Cholesky decomposition, where

$$Q = L^T L = \frac{1}{2}\begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 1 & 3 & 0 \end{bmatrix}\begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}$$

and so we can use the factorisation

$$x^T Q x = x^T L^T L x = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_3)^2 + \frac{1}{2}(x_2^2 + 3x_3)^2 \geq 0$$

A simple example of the use of Linear Matrix Inequality algorithms is to find P which meets the criteria in Theorem 2.12.

The S-Procedure can be used to determine whether a Linear Matrix Inequality holds on a specific domain. It is often useful to know whether

$$x^T P x \geq 0 \tag{2.28}$$

for all x such that $x^T R_i x \leq 0$, $i = 1, \ldots, m$. If there exist $\tau_1, \ldots, \tau_m \geq 0$ such that

$$x^T (P + \tau_i R_i) x \geq 0 \tag{2.29}$$

for all $x \in \mathbb{R}^n$ then (2.28) holds as $x^T P x \geq -\tau_i x^T R_i x \geq 0$ for all $x^T R_i x \leq 0$. In this method, the constants $\tau = (\tau_1, \ldots, \tau_m)$ are called multipliers, and the problem is a Linear Matrix Inequality in τ.

Example 2.20 *If we wish to determine whether $3 - x^2$ is non-negative between $-1 \leq x \leq 1$, then we can test whether there exists $\lambda \geq 0$ such that*

$$3 - x^2 + \lambda(x - 1)(x + 1) \geq 0 \tag{2.30}$$

For this case

$$P = \begin{pmatrix} -1 & 0 \\ 0 & 3 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.31}$$

and the condition holds for $\lambda = 2$.

2.6 Absolute Stability

Absolute Stability theory allows the study of nonlinear system stability using linear system tools. In this framework, a system is decomposed into a linear system with nonlinear feedback, where the nonlinearity is in a sector bounded by two linear functions. Stability criteria are then found using two choices of Lyapunov function, giving the circle and Popov criteria [41]. For simple systems this allows the stability criteria to be analysed geometrically [41]. Efficient computational methods have also been developed for the absolute stability framework using linear matrix inequalities [20, 34]. Similarly, the framework of Integral Quadratic Constraints has used efficient computational methods for linear matrix inequalities to study nonlinear systems [60].

Absolute Stability considers a linear time-invariant system with a memoryless, nonlinear, time-varying feedback, called the Lur'e problem. This is given by the set of equations

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \\ u &= \phi(y, t) \end{aligned} \tag{2.32}$$

where $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}^p$, (A, B) is controllable, (A, C) is observable, and $\phi : \mathbb{R}^p \times \mathbb{R}^+ \to \mathbb{R}^p$ is a memoryless, possibly time-varying nonlinearity, which is piecewise continuous in t and locally Lipschitz in y. It is also possible to set $y = Cx + Du$, but this is not studied in this thesis.

This can be written in the more compact form

$$\dot{x} = Ax + B\phi(Cx, t) \tag{2.33}$$

A sector can be given the following definition.

Definition 2.21 *A memoryless function $\phi : \mathbb{R}^p \times \mathbb{R}^+ \to \mathbb{R}^p$ is said to belong to the sector*

- $[K_1, K_2]$ *if* $[\phi(y,t) - K_1 y]^T [\phi(y,t) - K_2 y] \leq 0$.
- $[K_1, \infty]$ *if* $y^T[\phi(y,t) - K_1 y] \geq 0$.

It can be noted that there are special cases of the above function for $K_1 = 0$, where the sector $[0, \infty]$ is defined by $y^T \phi(y,t) \geq 0$.

Absolute Stability can be implemented in an LMI framework by considering the quadratic problem in (x, ϕ) as an LMI problem.

The circle criteria is obtained using a quadratic Lyapunov function

$$V = x^T P x \tag{2.34}$$

The Lyapunov function is positive definite if $P \succ 0$. For the case where $K_1 = 0$, the condition $\dot{V} \leq 0$ for ϕ on a sector becomes $\dot{V} - \tau\phi^T[\phi - K_2 y] \leq 0$, which results in

$$\begin{pmatrix} x^T & \phi^T \end{pmatrix} \begin{pmatrix} PA + A^T P + \varepsilon P & PB + \tau C^T K_2^T \\ B^T P + \tau K_2 C & -2\tau I \end{pmatrix} \begin{pmatrix} x \\ \phi \end{pmatrix} \leq 0 \tag{2.35}$$

For the Popov Criteria, the Lyapunov function candidate is $V = x^T P x + \lambda \int_o^x \phi(y) dy$. This results in the condition $P \succ 0$ and

$$\begin{pmatrix} x^T & \phi^T \end{pmatrix} \begin{pmatrix} PA + A^T P + \varepsilon P & PB + \lambda A^T C^T + \tau C^T K_2^T \\ B^T P + \lambda CA + \tau K_2 C & \lambda(CB + B^T C^T) - 2\tau I \end{pmatrix} \begin{pmatrix} x \\ \phi \end{pmatrix} \leq 0 \tag{2.36}$$

Example 2.22 *For the pendulum example, we have*

$$A = \begin{pmatrix} -\gamma & -\frac{1}{2}b \\ 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, C = \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{2.37}$$

with $K_2 = \frac{1}{2}b$. *The sector is valid in the domain* $-\pi/2 \leq x_2 \leq \pi/2$. *For* $\gamma = 1$ *and* $b = 1$, *then setting* $\varepsilon = 0.01$, *we can algorithmically find*

$$P = \begin{pmatrix} 1.0274 & 0.6849 \\ 0.6849 & 0.5137 \end{pmatrix} \tag{2.38}$$

which satisfies the circle criteria constraints, and so the system is absolutely stable.

A region of attraction estimate can also be found using the Lyapunov function [41].

2.7 Sum of Squares

A problem which arises in several areas of mathematics and engineering is checking whether a nonlinear function is non-negative [68] (see [25] for a review), i.e. whether

$$p(x) = p(x_1, \ldots, x_N) \geq 0 \tag{2.39}$$

This is often difficult to check directly. However, if $p(x)$ is a polynomial then we can relax this condition by the simpler, sufficient condition of whether a function is a sum of squares, such that

$$p(x) = \sum_i g_i(x)^2 \tag{2.40}$$

For example, $p(x) = (x_1 - 2x_2^2)^2 + (x_1 - x_2)^2$ is a Sum of Squares.

Testing whether a polynomial is a Sum of Squares can be converted into a Linear Matrix Inequality using the Gram Matrix Method [26], for which efficient computational methods exist [68].

Theorem 2.23 *A polynomial $p \in \mathbb{R}[x]$ of order $2d$ is a sum of squares if and only if there exists $Q \succeq 0$ such that*

$$p(x) = Z(x)^T Q Z(x) \tag{2.41}$$

where Z(x) is a vector of all monomials in x up to order d.

Proof. To show $\Rightarrow$, we use a Cholesky decomposition $Q = L^T L$ and set $g(x) = LZ(x)$ such that

$$p(x) = Z(x)^T Q Z(x) = Z(x)^T L^T L Z(x) = g(x)^T g(x) = \sum_i g_i(x)^2 \tag{2.42}$$

To show $\Leftarrow$, we use the reverse order, where we can find a constant matrix L such that $g(x) = LZ(x)$, where $Z(x)$ is a vector of monomials spanning polynomial vector $g(x)$. This gives

$$p(x) = \sum_i g_i(x)^2 = g(x)^T g(x) = Z(x)^T L^T L Z(x) = Z(x)^T Q Z(x) \tag{2.43}$$

where $Q = L^T L \succeq 0$. ■

We next give an example of this transformation to find a Sum of Squares polynomial.

Example 2.24 *Polynomial SOS [68]*

It is possible to test whether $p(x)$ is a polynomial SOS using the Gram Matrix Method, where

$$p(x_1, x_2) = 2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4 \tag{2.44}$$

Using $Z(x) = Z(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2)$ in (2.41) results in

$$Q = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

As seen in Example 2.19, $Q \succeq 0$ for $\lambda = -3$. Using a Cholesky decomposition $Q = L^T L$

then setting $g(x) = LZ(x)$ *obtains*

$$p(x) = g(x)^T g(x) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2$$

Implementing Sum of Squares can be completed using toolboxes such as SOSTOOLS [71].

Symmetry [27] and Sparsity [22, 43, 53, 87] can be used to reduce the complexity of calculations for reasons such as improving scalability, which will be discussed further in Chapter 4.

Sum of squares tests whether a polynomial is globally non-negative, however, we often wish to determine whether a polynomial is non-negative on a non-global domain. For this, a generalisation of the S-procedure can be used.

Theorem 2.25 *Generalised S-Procedure*

Given polynomials $p(x)$ *and* $a(x)$*, if there exists a vector of SOS polynomials* $s_1(x)$ *such that*

$$p(x) + s_1(x)^T a(x) \text{ is SOS} \tag{2.45}$$

then $p(x) \geq 0$ *for* $a(x) \leq 0$*.*

Proof. We have that $p(x) \geq -s_1(x)^T a(x) \geq 0$ for $a(x) \leq 0$. ■

Example 2.26 *To determine whether* $3 - x^4$ *is non-negative between* $-1 \leq x \leq 1$*, we can test whether there exists sum of squares* s_1 *such that*

$$3 - x^4 + s_1(x-1)(x+1) \text{ is SOS} \tag{2.46}$$

In this case $a(x) = (x-1)(x+1) \leq 0$ *for* $-1 \leq x \leq 1$*. For the sum of squares* $s_1 = 2(x^2+1)$ *we have*

$$3 - x^4 + 2(x^2+1)(x-1)(x+1) = x^4 + 1 \text{ is SOS} \tag{2.47}$$

Therefore $p(x) \geq 0$ *for all* $-1 \leq x \leq 1$*.*

2.7.1 Lyapunov theory and SOS Programming

Sum of Squares programming is a more recent computational method which has proved useful for finding Lyapunov functions for polynomial systems [25, 66, 68]. This technique has been shown to be useful for stability, region of attraction and robustness analysis for nonlinear systems, particularly further away from the equilibrium point [25, 68].

Theorem 2.27 *Let $x = 0$ be an equilibrium point for (2.3), for the case where $f(x), a(x)$ are polynomial vector fields, and $D = \{x \in \mathbb{R}^n | a(x) \leq 0\}$ is a domain containing a neighbourhood of $x = 0$. Let $V : D \to \mathbb{R}$ be a polynomial function, let ϕ_1, ϕ_2 be positive definite polynomial functions and let s_1 be SOS polynomials such that*

$$V(x) - \phi_1(x) \text{ is SOS} \tag{2.48}$$

$$\dot{V}(x) + s_1(x)^T a(x) + \phi_2(x) \text{ is SOS} \tag{2.49}$$

Then $V(x)$ is called a Lyapunov function and $x = 0$ in (2.3) is asymptotically stable.

A region of attraction estimate can also be found using this technique.

2.8 Conclusion

In this chapter, a background on nonlinear system theory was provided, which is used in subsequent chapters. ODEs and Lyapunov theory was presented for both the autonomous and time-varying cases. Absolute Stability theory, Linear Matrix Inequalities and Sum of Squares programming were also introduced, which are frameworks combining theory and computational methods and allow Lyapunov theory to be implemented algorithmically.

Chapter 3

Generalised Absolute Stability and Sum of Squares

3.1 Introduction

This chapter introduces a general framework for analysing nonlinear systems which are uncertain or time-varying, as well as systems with non-polynomial or higher order nonlinearities. This framework uses absolute stability theory and sum of squares programming for stability and region of attraction analysis. The technique decomposes a vector field into a system with a polynomial vector field in feedback with a nonlinear memoryless term, which is contained in a generalised sector inequality with polynomial bounds. This decomposition can be used to model uncertainty in the nonlinearity or to bound difficult-to-analyse terms by simpler polynomial functions, such as time varying, non-polynomial or higher order nonlinearities. Conditions for stability and regions of attraction are found using polynomial and Lur'e type Lyapunov functions, which generalise those used for the derivation of the multivariable circle and Popov criteria in classical absolute stability. The technique extends both absolute stability theory and the applicability of sum of squares programming. The usefulness of the technique is demonstrated with illustrative examples.

As discussed in Chapter 2, Absolute Stability is a traditional systems and control technique for dealing with stability of nonlinear systems [41], [29], [51]. In absolute stability, a nonlinear system is decomposed into a linear system with non-linear memoryless feedback, also known as the Lur'e problem. A quadratic inequality, otherwise known as a sector, is used to bound the non-linear feedback between two linear bounds, which allows frequency domain linear systems tools to be applied [41], [29], [51]. More recently, absolute stability conditions have been formulated in terms of LMIs (Linear Matrix Inequalities) [34], [39], [70], [81], due to the development of computationally efficient methods to solve them. Another technique with a similar decomposition uses Integral Quadratic Constraints to bound the nonlinear feedback [60].

As discussed in Chapter 2, Sum of Squares (SOS) programming, a technique for testing non-negativity of polynomial functions, can be used to implement systems analysis tools for systems with polynomial or rational vector fields [68], [69] (See [25] for a review). Moreover, a number of methods have been proposed to enable SOS to treat systems with non-polynomial vector fields. SOS programming can be applied to non-polynomial vector fields recast as polynomial systems with constraints [66], [67]. These systems can be analysed using polynomial equality, inequality and integral constraints on the system states and 'auxiliary variables', which represent inputs, non-polynomial functions and uncertain parameters [66]. Non-polynomial systems can also be approximated by a Taylor series, with treatment made rigorous by analysing error bounds using uncertain parameters [23, 24]. A class of non-polynomial or uncertain systems can be treated using SOS without auxiliary variables [84].

To generalise absolute stability, in this chapter we decompose the system into a polynomial or rational system and a nonlinear, memoryless feedback, which we describe as Lur'e type feedback. Generalised sector inequalities are used to bound the nonlinear feedback with polynomial functions. This compares with previous generalised sector conditions, which bound the nonlinear feedback with piecewise linear, concave/convex functions [38]. Conditions in the theorems can be tested using SOS programming in terms of LMIs, and so

the methodology presented in this chapter can be viewed as building upon the foundation of classical absolute stability with LMI conditions. The generalised sector condition allows less conservative bounds 'further away' from the equilibrium point (regionally or globally) than a classical sector condition, by which we mean that the error in the bounds or the uncertainty is smaller. The less conservative bounds reduce the conservativeness of region of attraction estimates or global stability conditions.

We treat SOS of non-polynomial or uncertain functions by converting 'functional' sum of squares problems to polynomial sum of squares. This method differs from many other SOS methods for non-polynomial functions, in that we treat the functions as independent variables, which are then related to existing variables with equality and inequality constraints [66]. In particular, we introduce the use of a generalised sector condition with polynomial bounds as an inequality constraint, which only becomes possible by treating the functions as independent variables. The alternative method of recasting also uses a change of variables together with equalities and inequalities [67], but does so at the expense of introducing higher order differential equations as well as inequality and equality constraints. Absolute stability generalises the Taylor series techniques introduced in [23, 24], as the Taylor series with error bounds can be used for the bounds in the generalised sector. The new sector can also allow us to use techniques other than Taylor series, which may be more suitable. The resulting conditions can also be viewed as a generalisation of the bounds proposed in [84], allowing more practical bounds to be used in many systems.

Section 3.2 discusses 'functional' SOS and introduces Generalised Sectors with polynomial bounds, which allow SOS programming to be applied to non-polynomial or uncertain functions. In Section 3.3 the framework of absolute stability is generalised and then used together with polynomial and Lur'e type Lyapunov functions to find stability conditions. Section 3.4 extends the stability analysis methodology to finding region of attraction estimates. Section 3.5 illustrates the usefulness of the techniques through their application to a range of examples.

3.2 Functional Sum of Squares and Generalised Sectors

In this section we study functional sum of squares, which are sum of squares of non-polynomial or uncertain functions. We give conditions under which they can be studied with polynomial sum of squares techniques by treating the functions as independent variables. This section introduces generalised sectors with polynomial bounds, which was first introduced in [38] for piecewise linear, convex/concave bounds, and then combines this with an expansion of the previous concept of using auxiliary variables in SOS programming to describe non-polynomial functions or parameters [68],[66].

3.2.1 Functional Sum of Squares

In this section we look at the problem of determining whether a function $p(x, \psi(x)) \geq 0$ for all $x \in D_x \subset \mathbb{R}^n$ where $\psi : D_x \to \mathbb{R}^p$ is a function of x. One method to test this is to determine the relaxed condition of whether $p(x, \psi(x))$ is SOS. If

$$p(x, \psi(x)) = \sum_{i=1}^{m} g_i(x, \psi(x))^2 \tag{3.1}$$

then $p(x, \psi(x))$ is SOS and so $p(x, \psi(x)) \geq 0$ for $x \in D_x$, where each g_i can be a non-polynomial function. We refer to this as a functional SOS rather than a polynomial SOS. If the function $p(x, \psi(x))$ can be defined globally (i.e. $D_x = \mathbb{R}^n$), then the 'functional' SOS implies global non-negativity. For this case, Ψ may be discontinuous, however, in later section where Ψ defines a vector field then further continuity assumptions are required.

Now if $p(x, \psi)$ is a polynomial in (x, ψ) then we can make a further relaxation by treating ψ as an independent variable with $\psi \in \mathbb{R}^p$ and determine whether $p(x, \psi)$ is a polynomial SOS, where

$$p(x, \psi) = \sum_{i=1}^{m} g_i(x, \psi)^2 \tag{3.2}$$

and $g_i(x,\psi)$ is a polynomial function in x and ψ. If $p(x,\psi)$ is a polynomial of degree $2d$ in x and ψ then we can reduce the test of whether $p(x,\psi)$ is a SOS into a Linear Matrix Inequality using the Gram Matrix Method [26]. We let

$$p(x,\psi) = Z(x,\psi)^T Q Z(x,\psi) \tag{3.3}$$

where $Z(x,\psi) \in \mathbb{R}^m$ is a vector of monomials of x and ψ of degree less than or equal to d and $Q = Q(\lambda) \in \mathbb{R}^{m\times m}$ is affinely parameterised by $\lambda \in \mathbb{R}^\kappa$ as it is a non-unique representation due to the coupling of elements in $Z(x,\psi)$. Then $p(x,\psi)$ is a SOS in x and ψ if there exists a choice of parameters λ such that $Q \succeq 0$.

Example 3.1 *This example is used to illustrate functional SOS reduced to polynomial SOS. It is a modification of an example in [68] originally used to illustrate polynomial SOS. We wish to study whether $p(x)$ is a functional SOS, where*

$$p(x) = 2x^4 + 2x^3 \sin x - x^2 \sin^2 x + 5\sin^4 x \tag{3.4}$$

By setting $\psi(x) = \sin x$ and treating p as a polynomial in x and ψ, this becomes

$$p(x,\psi) = 2x^4 + 2x^3\psi - x^2\psi^2 + 5\psi^4 \tag{3.5}$$

Using $Z(x,\psi) = (x^2, \psi^2, x\psi)$ in (3.3) results in

$$Q = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix} + \lambda \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}.$$

$Q \succeq 0$ for $\lambda = -3$ and a Choleszky decomposition obtains

$$\begin{aligned} p &= \frac{1}{2}(2x^2 - 3\psi^2 + x\psi)^2 + \frac{1}{2}(\psi^2 + 3x\psi)^2 \\ &= \frac{1}{2}(2x^2 - 3\sin^2 x + x\sin x)^2 + \frac{1}{2}(\sin^2 x + 3x\sin x)^2 \end{aligned}$$

In the above description we show that p is a SOS for all $\psi : \mathbb{R}^n \to \mathbb{R}^p$, and hence for $\psi(x) = \sin x$. However, we often have more information about the function $\psi(x)$, which we can use to obtain less conservative conditions. Indeed, we can relate $\psi(x)$ to x through polynomial equalities and inequalities, which can be used with a generalised S-procedure to obtain less conservative SOS conditions for the non-negativity of $p(x, \psi(x))$.

Theorem 3.2 *Let $p(x, \psi)$ be a polynomial in x and ψ, $a(x) \in \mathbb{R}^b[x]$ and $D_x = \{x \in \mathbb{R}^n | a(x) \preceq 0\}$. Let $r(x, \psi) \in \mathbb{R}^c[x, \psi]$ and let $r(x, \psi(x)) \preceq 0$ for all $x \in D_x$. Let $s_{1i}(x, \psi), s_{2k}(x, \psi)$ be polynomial SOS in (x, ψ) for $i = 1, \ldots, c$ and $k = 1, \ldots, b$. If*

$$p(x, \psi) + s_1(x, \psi)^T r(x, \psi) + s_2(x, \psi)^T a(x) \text{ is SOS in } (x, \psi) \tag{3.6}$$

then $p(x, \psi(x)) \geq 0$ for all $x \in D_x$.

Proof. $a(x), r(x, \psi(x)) \preceq 0$ for all $x \in D_x$ and so from (3.6)

$$p(x, \psi(x)) \geq -s_1(x, \psi(x))^T r(x, \psi(x)) - s_2(x, \psi(x))^T a(x) \geq 0$$

for all $x \in D_x$. ■

We could combine $a(x) \leq 0$ and $r(x, \psi) \leq 0$ into one inequality, but we separate them in order to clearly identify the inequality which defines the domain of x and the inequality which relates x to ψ on that domain. When (3.6) is placed in the form of an LMI then ψ and x are related directly through the inequality $r(x, \psi) \leq 0$ as well the higher order equality relations as in Example 3.1.

Equation (3.6) can be tested using SOS programming techniques described above with toolboxes such as SOSTOOLS [71].

3.2.2 Generalised Sector Conditions

We next introduce a generalised sector constraint with polynomial or rational function bounds. Let $\phi : D_y \to D_u$, $D_y \subset \mathbb{R}^r, D_u \subset \mathbb{R}^p$, where $\phi(y)^T = (\phi_1(y)^T, \ldots, \phi_s(y)^T)$, $\phi_i : D_y \to \mathbb{R}^{q_i}$ for $i = 1, \ldots, s$ and $\sum_i q_i = p$. We place each function $\phi_k(y)$ in a generalised sector

$$[\phi_k(y) - \alpha_k(y)]^T [\phi_k(y) - \beta_k(y)] \leq 0 \tag{3.7}$$

for all $y \in D_y$, where $\alpha_k(y), \beta_k(y) \in \mathbb{R}^{q_i}[y]$ are polynomial vector functions. To ensure that (3.7) is well defined, we require that the intersection of the different constraints is non-empty for all $y \in D_y$.

We can use this framework to treat both uncertain parameters and uncertain functions if the uncertain parameters have an upper and lower bound. In this framework, a parameter $p \in \mathbb{R}$ which is bounded between $a \leq p \leq b$ will have $\phi_k = p$, $\alpha_k = a$, $\beta_k = b$ and so $[\phi_k - a][\phi_k - b] \leq 0$.

We can also use a generalised sector to simplify application of rational bounds

$$[A_k(y)\phi_k(y) - \alpha_k(y)]^T [B_k(y)\phi_k(y) - \beta_k(y)] \leq 0 \tag{3.8}$$

for all $y \in D_y$, where $A_k, B_k \in \mathbb{R}^{q_i \times q_i}[y]$ are diagonal matrices and the diagonals of A, B are positive for all $y \in D_y$. The elements of $A(y)$ and $B(y)$ can represent the denominator of the bounds and α, β can represent the numerator. For the one dimensional case this can be seen to be equivalent to

$$\left[\phi_k(y) - \frac{\alpha_k(y)}{A_k(y)}\right] \left[\phi_k(y) - \frac{\beta_k(y)}{B_k(y)}\right] \leq 0 \tag{3.9}$$

but (3.8) has the advantage of being in polynomial form.

The generalised sector has a number of advantages for SOS programming and absolute stability. Firstly, it is a simple but powerful method for bounding nonlinear functions.

It is also practical for SOS programming as it has an even leading order power in ϕ_k and, if the order of β and α are the same, it also has an even leading order power in y. Intuitively, even leading orders may help computationally in finding SOS polynomials and multipliers, as SOS polynomials have even leading orders. Finally, the sector bounds are defined such that there is no uncertainty at $x = 0$. When applied to vector fields in nonlinear systems, such as in Section 3.3, this definition ensures that the system has an equilibrium point at $x = 0$ for all functions in the sector.

The tradeoff when using a generalised sector in the form (3.7) is that the order of the resulting polynomial is higher than two inequalities each with a single bound. This issue can often be resolved by using appropriate bounds for the non-polynomial or uncertain function as can be seen in the following example.

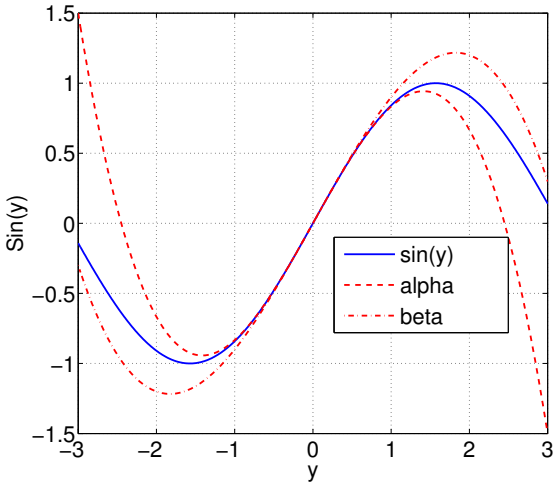


Figure 3.1: Generalised Sector for $\sin(y)$ for $-3 \leq y \leq 3$. $\alpha = y - \frac{1}{6}y^3$ and $\beta = y - \frac{1}{10}y^3$

Example 3.3 *We wish to find polynomial bounds for the function* $\sin(y)$*. We can use the Taylor series as a starting point for the sector bounds, where* $\sin y = y - \frac{1}{3!}y^3 + \frac{1}{5!}y^5 - \frac{1}{7!}y^7 + O(y^9)$ *and so we can use* $\alpha(y) = y - \frac{1}{6}y^3$ *and* $\beta(y) = y - \frac{1}{10}y^3$ *for* $-\pi \leq y \leq \pi$*. Using these bounds, which are* 3^{rd} *order, simplifies analysis over the related Taylor series approximation with an error bound, which is* 5^{th} *order. The difference between bounds is tighter near* $y = 0$*, but other bounds can be used for which the error is more uniform.*

Alternatively, we can set $\sin y = y + \phi(y)y^3$*. If we treat* ϕ *as an uncertain parameter*

bounded by constants then this is equivalent to a 1st order Taylor series, where ϕy^3 represents the error term. An extension of the alternative bounds for ϕ is $\alpha(y) = -\frac{1}{3!} + \frac{1}{5!}y^2$ and $\beta(y) = -\frac{1}{3!} + \frac{7}{10}\frac{1}{5!}y^2$ for $-\pi \leq y \leq \pi$.

In the above example, ad hoc methods based upon the Taylor series were used. The best polynomial and rational bounds to use for generalised sectors is an interesting future direction of research.

We can represent $r(x, \psi) \leq 0$ in Theorem 3.2 as a generalised sector if we set $y = h(x)$ and $\psi(x) = \phi(h(x))$, where $h : D_x \to D_y$ is a polynomial function. Introducing y allows us to simplify equalities and inequalities e.g. for $\sin(x_1^2 - x_2^2)$ we can set $y = x_1^2 - x_2^2$ and find simpler bounds for $\sin y$.

The generalised sector bounds based on polynomials have advantages of smoothness, nonlinearity, and no requirement for convexity or concavity when compared to piecewise linear, convex/concave bounds [38]. The smoothness allows the bound to be described by one 'piece', while the convexity/concavity relaxation allows tighter sector bounds for functions which are not convex or concave. When applied to vector fields in nonlinear systems, some cases can be indeterminate when using linear bounds, but can be determined using polynomial bounds e.g. a linearised system with eigenvalues that have zero real parts.

3.2.3 Generalised Sectors for Integral Inequalities

We can combine generalised sector conditions with integral inequalities, where we can find generalised sector bounds for an integral by finding bounds for its integrand, such as for

$$\Phi(y) = \int_0^y \gamma(z)dz \tag{3.10}$$

where $\gamma(z)$ and $\Phi(y)$ are scalar functions. We use the integrand bounds $y\rho_1(y) \leq y\gamma(y) \leq y\rho_2(y)$ for all $y \in D_y$, where ρ_1, ρ_2 are polynomial. This leads to

$$\int_0^y \rho_1(z)dz \leq \Phi(y) \leq \int_0^y \rho_2(z)dz. \tag{3.11}$$

which can be used for bounding $\Phi(y)$ in (3.7) as the integral of a polynomial is also polynomial. This is a useful technique as polynomials are typically much simpler to integrate than $\gamma(z)$, and this technique can be easily generalised for a non-scalar function $\gamma(z)$.

3.3 Generalised Absolute Stability

In this section we use the techniques developed in Section 3.2 to generalise the concept of absolute stability.

3.3.1 Generalised Absolute Stability Definitions

In order to deal with models that have nonlinearities and uncertainty, we study systems which can be decomposed into a polynomial or rational system with a nonlinear or uncertain feedback:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x) \\ u &= \phi(t,y) \end{aligned} \tag{3.12}$$

where $x \in D_x \subset \mathbb{R}^n$, $u \in D_u \subset \mathbb{R}^p$, $y \in D_y \subset \mathbb{R}^r$, $f : D_x \times D_u \to \mathbb{R}^n$ is a linear, polynomial or rational function, $h : D_x \to D_y$ is a polynomial function, $\phi : \mathbb{R}^+ \times D_y \to D_u$ is a nonlinear function and $x(0) = x_0 \in D_x$. We assume that $f(0, \phi(t, h(0))) = 0$, D_x contains a neighbourhood of the origin, and the system (3.12) is locally Lipschitz in x and continuous in t for all $x \in D_x$ and $t \geq 0$. If f is a rational function then we call its least common denominator f_d.

We now use a generalised sector for time varying functions $\phi(t,y)^T = (\phi_1(t,y)^T, \ldots, \phi_s(t,y)^T)$, where $\phi_k : \mathbb{R}^+ \times D_y \to \mathbb{R}^{q_k}$ for $k = 1, \ldots, s$ and $\sum_k q_k = p$. We place each function $\phi_k(t,y)$ in a polynomial generalised sector

$$r_k(y,\phi) = [\phi_k(t,y) - \alpha_k(y)]^T [\phi_k(t,y) - \beta_k(y)] \leq 0 \tag{3.13}$$

or a rational generalised sector

$$\begin{aligned} &r_k(y,\phi) \\ &= [A_k(y)\phi_k(t,y) - \alpha_k(y)]^T [B_k(y)\phi_k(t,y) - \beta_k(y)] \leq 0 \end{aligned} \tag{3.14}$$

for all $y \in D_y$, $t \geq 0$ and $k = 1, \ldots, s$, where $\alpha_k(y), \beta_k(y) \in \mathbb{R}^{q_k}[y]$ are polynomial vector functions in y and for (3.13) $A_k, B_k \in \mathbb{R}^{q_k \times q_k}[y]$ are diagonal matrices where the diagonals are positive for all $y \in D_y$. We assume that the intersection of constraints is non-empty for all $y \in D_y$. We also require that $f(0, \phi(t, \bar{y})) = 0$ for all $\bar{y} = h(0)$ meeting the condition (3.13) or (3.14), which for $q_k = 1$ is the same as requiring $f(0, \alpha(\bar{y})) = f(0, \beta(\bar{y})) = 0$ for (3.13) or $f(0, A^{-1}(\bar{y})\alpha(\bar{y})) = f(0, B^{-1}(\bar{y})\beta(\bar{y})) = 0$ for (3.14), where $A = \text{diag}(A_1, \ldots, A_s)$, $B = \text{diag}(B_1, \ldots, B_s)$.

The assumptions on ϕ are built into the requirement that System (3.12) is well defined (i.e. locally Lipschitz and continuous in t) and that $f = 0$ when $x = 0$. For the cases considered in this thesis, the assumption on the non-empty intersection of constraints is not restrictive, and is included so as not to allow the degenerate case.

We can use this framework to treat both uncertain functions and uncertain polynomials with upper and lower bounds by using ϕ_k to represent uncertain parameters and setting α_k, β_k as constants.

Without loss of generality, the equilibrium point is set at $x = 0$ as a non-zero equilibrium point can be translated to the origin. An uncertain equilibrium point can be represented in this framework using uncertain parameters, which represent the uncertainty in the translation. An alternative method to represent uncertain, non-zero equilibrium points

is to use equilibrium-independent passivity [35]. In the case of equilibrium-independent passivity, the equilibrium can be at an arbitrary input and output, rather than at zero input and zero output, and the passivity property holds about this point.

Remark 3.4 *In this chapter we assume that the feedback is memoryless, but we could also use the same framework to impose integral polynomial constraints [66] on the feedback e.g. to account for unmodelled dynamics. This would allow the above theory to be more closely tied to IQCs [60].*

Example 3.5 *Consider the pendulum system*

$$\begin{aligned} \dot{x}_1 &= -\gamma x_1 - \sin(x_2) + p \\ \dot{x}_2 &= x_1 \end{aligned} \tag{3.15}$$

where $0 \leq p \leq 1$*, as the system is normalised. The equilibrium points occur at* $\overline{x}_1 = 0$ *and* $\sin(\overline{x}_2) = p$*. Setting* $z_1 = x_1 - \overline{x}_1$ *and* $z_2 = x_2 - \overline{x}_2$ *obtains the equations*

$$\begin{aligned} \dot{z}_1 &= -\gamma z_1 - \sin(z_2 + \overline{x}_2) + p \\ \dot{z}_2 &= z_1 \end{aligned} \tag{3.16}$$

There are multiple ways of defining the Lur'e type problem. As one case, it is possible to set

$$\begin{aligned} f_1 &= -\gamma z_1 - u \\ f_2 &= z_1 \\ y &= h(x_2) = z_2 \\ \phi(y) &= \sin(y + \overline{x}_2) - p \end{aligned} \tag{3.17}$$

so that $\phi(0) = 0$*. The difficulty with this is trying to find simple bounds for* ϕ*, as the bounds are dependent upon* p*. Instead, we use*

$$\sin(z_2 + \overline{x}_2) - \sin(\overline{x}_2) = q\sin(z_2) - p(1 - \cos(z_2))$$

where $q = \sqrt{1-p^2}$*, which leads to the form*

$$\begin{aligned} \dot{z}_1 &= -\gamma z_1 - q u_1 - p u_2 \\ \dot{z}_2 &= z_1 \\ y &= h(z) = z_2 \\ \phi_1(y) &= \sin(y), \phi_2(y) = 1 - \cos(y) \end{aligned} \tag{3.18}$$

This form is advantageous as the bounds for the non-polynomial function are not dependent on $\overline{x}_2$*. For the generalised sector,* $5^{th}/6^{th}$ *order bounds can be set*

$$z_2 - \frac{1}{6}z_2^3 + \frac{0.78}{5!}z_2^5 \leq \phi_1(z_2) \leq z_2 - \frac{1}{7}z_2^3 + \frac{0.52}{5!}z_2^5$$

$$\frac{1}{2}z_2^2 - \frac{1}{4!}z_2^4 + \frac{0.8}{6!}z_2^6 \leq \phi_2(z_2) \leq \frac{1}{2}z_2^2 - 0.9\frac{1}{4!}z_2^4 + \frac{0.56}{6!}z_2^6$$

If Example 3.5 had an uncertain equilibrium, this could be treated by setting p*, and hence* q*, an uncertain parameter.*

Once the system has been placed in the correct form, it is possible to analyse stability. To do this, some new definitions of stability are required for the new framework.

Definition 3.6 *We let* F_r *be the set of all functions* $\phi(t, y)$ *such that* $r(y, \phi(t, y)) \leq 0$ *for all* $y \in D_y$ *and all* $t \geq 0$*.*

For system (3.12) *where* ϕ_k *satisfies sector* (3.13) *or* (3.14) *on domain* D_x*, the system is 'absolutely stable on a finite domain' if the origin is uniformly asymptotically stable for all* $\phi \in F_r$*. It is 'absolutely stable' if* $D_x = \mathbb{R}^n$ *and the origin is globally uniformly asymptotically stable for all* $\phi \in F_r$*.*

This is equivalent to the classical absolute stability definition for the special case that the functions and sector bounds are linear.

3.3.2 Polynomial Lyapunov functions

We first generalise the use of the circle criterion for absolute stability. We use a Lyapunov function candidate $V(x)$ which is a polynomial of degree $2d$. The multivariable circle criterion can be obtained by setting $d = 1$ i.e. $V(x) = x^T Px$ is a quadratic function.

Theorem 3.7 *Let $a(x) \in \mathbb{R}^b[x]$, the domain D_x be defined by $D_x = \{x \in \mathbb{R}^n | a(x) \leq 0\}$, $V(x)$ be a polynomial where $V(0) = 0$, $s_1 \in \mathbb{R}^p[x, \phi], s_2 \in \mathbb{R}^b[x, \phi]$ be SOS, $\sigma_1(x)$, $\sigma_2(x)$ be positive definite polynomial functions and define*

$$p_1(x, \phi) = -\nabla V(x) f(x, \phi) - \sigma_2(x) + s_1(x, \phi)^T r(x, \phi) + s_2(x, \phi)^T a(x).$$

For $r(x, \phi) \leq 0$ defined either by (3.13) *or by* (3.14)*, if*

$$V(x) - \sigma_1(x) \text{ is SOS in } x \tag{3.19}$$

$$f_d(x, \phi) p_1(x, \phi) \text{ is SOS in } (x, \phi) \tag{3.20}$$

where $f_d = 1$ if $f(x, \phi)$ is polynomial, then (3.12) *is absolutely stable on a finite domain. Furthermore, it is absolutely stable if $D_x = \mathbb{R}^n$.*

Proof. We note that the polynomial case is the special case where $f_d = 1$ and $A = B = I$.

From (3.19) then $V(x) \geq \sigma_1(x)$ where $\sigma_1(x)$ is a positive definite function. As $V(0) = 0$ and $V(x)$ is not a function of t then there exists a positive define function $\sigma_3(x)$ such that $\sigma_1(x) \leq V(x) \leq \sigma_3(x)$.

As $f_d p_1$ is SOS and $f_d > 0$ for all $x \in D_x$ then

$$-\dot{V}(x) - \sigma_2(x) \geq -s_1(x, \phi) r(x, \phi) - s_2(x, \phi) a(x) \geq 0$$

for all $x \in D_x$ and all $\phi \in F_r$ and so $\dot{V}(x) \leq -\sigma_2(x)$ for all $x \in D_x$ and all $\phi \in F_r$. Thus by Theorem 2.15, $x = 0$ in System (3.12) is uniformly asymptotically stable for all $\phi \in F_r$.

If $D_x = R^n$ then $\sigma_1(x)$ is radially unbounded as it is a polynomial and so by Theorem 2.15, $x = 0$ in System (3.12) is globally uniformly asymptotically stable for all $\phi \in F_r$. ■

In the theorem above, f_d is the denominator of f, which is used to transform the problem to a polynomial SOS problem for a rational vector field.

Example 3.8 *Continuing Example 3.5, we look at the simplest case, taking the simplest case of $p = 0$ and the bounds as*

$$z_2 - \frac{1}{10}z_2^3 + \leq \phi_1(z_2) \leq z_2 - \frac{1}{6}z_2^3$$

For a polynomial Lyapunov function, for 2^{nd} order $a(x)$ and 4^{th} order $p(x)$, using SOSTOOLS obtains:

$$V(x) = 26.583z_1^2 + 22.582z_2z_1 + 13.684z_2^2$$

which is valid for the domain $-0.75\pi \leq z_2 \leq 0.75\pi$. The limit when expanding the domain D is the constraints on the domain for the sector, rather than constraints for V or $\dot{V}$.

3.3.3 Lur'e type Lyapunov functions

We next generalise the use of the Popov criterion for absolute stability. For the Lur'e type Lyapunov function we look at the case where $f(x, \phi)$ is polynomial, $\phi = \phi(y)$ is independent of time and $\phi_k = \phi_k(y_{i(k)})$, where $i(k)$ is a function of k. We use the Lyapunov function candidate

$$V(x) = \overline{V}(x) + \sum_{k=1}^{p} \gamma_k \Phi_k \tag{3.21}$$

where $\overline{V}(x)$ is polynomial of degree $2d$, γ_k are positive constants and

$$\Phi_k = \int_0^{y_{i(k)}} \phi_k(z)dz \tag{3.22}$$

where Φ_k is a scalar function. The multivariable Popov criterion can be obtained by setting $d = 1$, so $\overline{V}(x) = x^T Px$ is a quadratic function, and by setting $i(k) = k$.

By using a function which contains integrals of vector field terms, it is computationally easier to find Lyapunov function candidates $V(x)$ for which $\dot{V}(x) \leq 0$, typically the more complex condition when applying Lyapunov function theory. The disadvantage of this approach is that the function $V(x)$ itself becomes more complicated to analyse.

We use a generalised sector bound for both ϕ_k and its integral Φ_k, where

$$\xi_k(y_{i(k)}) \leq \Phi_k(y_{i(k)}) \leq \zeta_k(y_{i(k)}) \tag{3.23}$$

for all $y \in D_y$, all $\phi \in F_r$ and $k = 1, \ldots, p$. It can be noted that (3.23) holds for all $x \in D_x$ as $y = h(x)$ is required to map from D_x to D_y.

We next find generalised absolute stability conditions using the generalised Lur'e type Lyapunov function as well as by using a bound for the integral.

Theorem 3.9 *Let $a(x) \in \mathbb{R}^b[x]$, the domain D_x be defined by $D_x = \{x \in \mathbb{R}^n | a(x) \leq 0\}$, $\overline{V}(x)$ be a polynomial where $\overline{V}(0) = 0$, $s_1(x) \in \mathbb{R}^b[x], s_2 \in \mathbb{R}^p[x, \phi], s_3 \in \mathbb{R}^b[x, \phi]$ be SOS, $\sigma_1(x)$, $\sigma_2(x)$ be positive definite polynomial functions and define*

$$p_1(x) = \overline{V}(x) + \sum_k \gamma_k \xi_k(h_{i(k)}(x)) - \sigma_1(x) + s_1(x)^T a(x) \tag{3.24}$$

$$\begin{aligned} p_2(x, \phi) = &- \frac{\partial \overline{V}}{\partial x} f(x, \phi) - \sum_k \gamma_k \phi_k \frac{\partial h_{i(k)}}{\partial x} f(x, \phi) \\ &+ s_2(x, \phi)^T r(x, \phi) + s_3(x, \phi)^T a(x) - \sigma_2(x) \end{aligned} \tag{3.25}$$

For polynomial f and $r(x, \phi) \leq 0$ defined by (3.13)*, if*

$$p_1(x) \text{ and } p_2(x, \phi) \text{ are SOS} \tag{3.26}$$

then (3.12) *is absolutely stable on a finite domain.*

Proof. As p_1 is SOS then

$$\overline{V}(x) + \sum_k \gamma_k \xi_k(h_{i(k)}(x)) - \sigma_1(x) \geq -s_1(x)^T a(x) \geq 0$$

for all $x \in D_x$ and all $\phi \in F_r$. Thus $V(x) \geq \sigma_1(x)$ for all $x \in D_x$ and all $\phi \in F_r$ as $\Phi_k \geq \xi_k$. Also, $\overline{V}(0) = 0$, $\Phi_k(h_{i(k)}(0)) = 0$ and so $V(0) = 0$. As $V(0) = 0$ and $V(x)$ is not a function of t then there exists a positive definite function $\sigma_3(x)$ such that $\sigma_1(x) \leq V(x) \leq \sigma_3(x)$.

As p_2 is SOS then

$$\begin{aligned} &- \frac{\partial \overline{V}}{\partial x} f(x, \phi(x)) - \sum_k \gamma_k \phi_k \frac{\partial h_{i(k)}}{\partial x} f(x, \phi(x)) - \sigma_2(x) \\ &\geq -s_2(x, \phi(x))^T r(x, \phi(x)) - s_3(x, \phi(x))^T a(x) \geq 0 \end{aligned}$$

for all $x \in D_x$ and all $\phi \in F_r$. Thus

$$\begin{aligned} \dot{V}(x) &= \frac{\partial \overline{V}}{\partial x} \dot{x} + \sum_k \gamma_k \phi_k \dot{y}_{i(k)} \\ &= \frac{\partial \overline{V}}{\partial x} f(x, \phi) + \sum_k \gamma_k \phi_k \frac{\partial h_{i(k)}}{\partial x} f(x, \phi) \leq -\sigma_2(x) \end{aligned}$$

for all $x \in D_x$ and all $\phi \in F_r$. Therefore, by Theorem 2.15, $x = 0$ is uniformly asymptotically stable for all $\phi \in F_r$.

The proof of global uniform asymptotic stability is as per proof in Theorem 3.7. ■

We do not look at the case of rational bounds for the Lur'e type function as the integrated bounds will not necessarily be polynomial or rational.

Example 3.10 *Continuing Example 3.8 and using a Lur'e type Lypaunov function, for a 2^{nd} order $a(x)$ and a 4^{th} order $p(x)$ the returned Lyapunov function is*

$$\sigma = 13.072 z_1^2 + 8.9604 z_1 z_2 + 4.7886 z_2^2$$

$$\gamma = 13.18$$

This is valid in the domain $-0.75\pi \leq z_2 \leq 0.75\pi$. *It can be noted that*

$$\sigma + \xi_1(z) = 13.072z_1^2 + 8.9604z_1z_2 + 11.38x_2^2 - 0.55z_2^4$$

which is positive definite on D.

As for Example 3.8, the limit is the domain for the sector, rather than for V *or* $\dot{V}$.

3.4 Region of Attraction

In this section we provide a method for determining the region of attraction for systems with uncertain or non-polynomial vector fields using the types of Lyapunov functions introduced in this chapter. Region of Attraction (ROA) estimates for polynomial systems can be found using Lyapunov functions and SOS programming [68],[65],[83],[82],[85]. The problem of finding the largest region of attraction estimate for polynomial systems using SOS leads to a Bilinear Matrix Inequality, as the coefficients need to be determined for both polynomial representing the domain and its SOS multiplier [83],[82]. Thus find the largest region of attraction is non-convex.

In generalised absolute stability, a region of attraction estimate can still be found with standard SOS techniques, with the main difference that the region D_x can only be modified as long as $\phi_k(y)$ is contained on the defined sector for the finite domain. Also, it is common for the Lur'e type Lyapunov function not to be radially unbounded or monotonically increasing and so more care is required when analysing sub-level sets.

In this section we illustrate how to find a region of attraction estimate in the framework of generalised absolute stability. To illustrate this, we use a simple technique based upon [65].

Recall that F_r is the set of all functions $\phi(t, y)$ such that $r(y, \phi(t, y)) \leq 0$ for all $y \in D_y$ and all $t \geq 0$. We let $\pi(x_0, t_0, \phi, t)$ be a solution to system (3.12) with initial condition

$x(t_0) = x_0$ and a particular function $\phi(t, y) \in F_r$.

Definition 3.11 *The (robust) region of attraction R_a for $t_0 = 0$ is defined as $R_a = \{x_0 \in D_x | \pi(x_0, 0, \phi, t) \to 0$ as $t \to \infty$, for all $\phi(t, y) \in F_r\}$. A positively invariant subset of R_a which contains the origin is called a region of attraction estimate.*

We next give conditions in terms of SOS using polynomial and Lur'e type Lyapunov functions so that the conditions are computable.

3.4.1 ROA using Polynomial Lyapunov function

We first find ROA estimates using Polynomial Lyapunov functions, where we use the definition $\Omega_c = \{x \in D_x | V(x) \leq c\}$.

Theorem 3.12 *If the conditions for Theorem 3.7 are met and there exists $c, \varepsilon > 0$, polynomial $p_j(x)$, and integers r_j such that*

$$(x^T x)^{r_j}(V(x) - c - \varepsilon) + p_j(x)a_j(x) \text{ is SOS} \tag{3.27}$$

for $j = 1, \ldots, b$ then Ω_c is a region of attraction estimate.

Proof. We let $M = \{x \in D_x | V(x) \leq c + \varepsilon\}$ in Theorem 2.15. The condition $V(x) - \sigma_1(x)$ is SOS implies that M is compact as $\sigma_1(x)$ is radially unbounded. Also, for $x \neq 0$ then (3.27) implies $V(x) \geq c + \varepsilon$ for $a_j(x) = 0$ and thus $V(x) \geq c + \varepsilon$ for ∂M. Setting $\bar{\sigma}_1(x) = \bar{\sigma}_3(x) = V(x)$, where $V(x)$ is positive definite, then Theorem 2.15 implies Ω_c is an ROA subset. As Ω_c is positively invariant then Ω_c is also an ROA estimate. ■

We treat the set defined by $a_1(x) \leq 0$ as an arbitrary set whose area we wish to maximise while using $(a_2(x), \ldots, a_b(x)) \leq 0$ to describe the region in which the generalised sector is valid, after removing redundant constraints.

Using a simple, iterative method, a first step is to choose $a_1(x) = x^T x - \eta$ and find V, s_1, s_2 such that η is maximised subject to the conditions in Theorem 3.7. This is a Bilinear

Matrix Inequality, for which the simplest method is to determine V, s_1, s_2 for incrementally increasing values of η until the conditions in Theorem 3.7 cannot be met. A second step is to similarly maximise c such the the condition in Theorem 3.12 holds. The two steps are then repeated for $a_1(x) = V_p(x) - \eta$, where V_p is the choice of $V(x)$ from the previous iteration. As the process is iterated, $V(x), s_1(x), s_2(x)$ are typically chosen of increasing orders to maximise the region of attraction estimate.

3.4.2 ROA using Lur'e type Lyapunov functions

We can also find a subset of the region of attraction for Lur'e type Lyapunov functions. If we find the Lur'e type Lyapunov function in (3.21) using Theorem 3.9 then we use the modified definition $\Omega_c = \{x \in D_x | \overline{V}(x) + \sum_k \gamma_k \Phi_k \leq c\}$.

Theorem 3.13 *Let*

$$\Omega_{\zeta,c} = \{x \in D_x | \overline{V}(x) + \sum_k \gamma_k \zeta_k(h_{i(k)}(x)) \leq c\} \tag{3.28}$$

If the conditions for Theorem 3.9 are met, D_x is bounded and if there exists $c, \varepsilon > 0$, polynomials $p_j(x)$ and integers r_j for $j = 1, \ldots, b$ such that

$$(x^T x)^{r_j}(\overline{V}(x) + \sum_k \gamma_k \xi_k(h_{i(k)}(x)) - c - \varepsilon) + p_j(x)a_j(x) \text{ is SOS} \tag{3.29}$$

then $\Omega_{\zeta,c}$ is a subset of the region of attraction.

Proof. In Theorem 2.15, we set $\bar{\sigma}_1(x) = \overline{V}(x) + \sum_k \gamma_k \xi_k$, $\bar{\sigma}_3(x) = \overline{V}(x) + \sum_k \gamma_k \zeta_k$ and $M = \{x \in D_x | \bar{\sigma}_1(x) \leq c + \varepsilon\}$. $\bar{\sigma}_1(x)$ and $\bar{\sigma}_3(x)$ are positive definite functions as $\bar{\sigma}_1(0) = \bar{\sigma}_3(0) = 0$ and $\bar{\sigma}_3(x) \geq \bar{\sigma}_1(x) \geq \sigma_1(x)$ for all $x \in D_x$. Bounded D_x implies that M is compact. For $x \neq 0$ then (3.29) implies $\overline{V}(x) + \sum_k \gamma_k \xi_k \geq c + \varepsilon$ for $a_j(x) = 0$ and thus $\bar{\sigma}_1(x) \geq c + \varepsilon$ for ∂M. Thus Theorem 2.15 implies $\Omega_{\zeta,c}$ is an ROA subset. ■

Example 3.14 *The technique for ROA analysis is illustrated using the system in Example 3.5.*

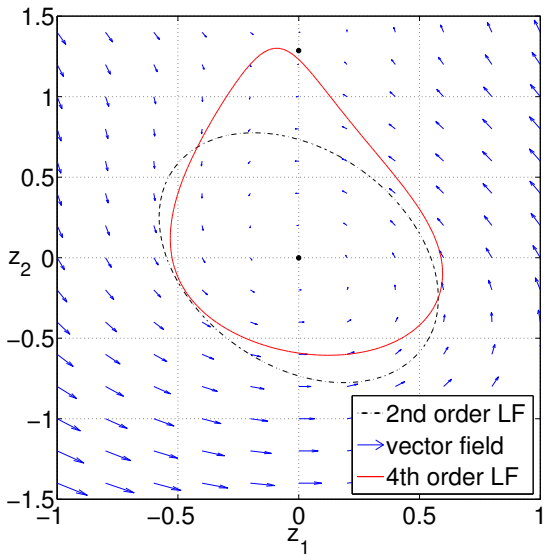


Figure 3.2: Region of Attraction estimates using Polynomial Lyapunov functions when $p = 0.8, \gamma = 0.4$. The stable equilibrium point is at $(0, 0)$ and an unstable equilibrium point is at $(0, 1.287)$. The vector field is described by Equation (3.16).

The respective bounds for the Lur'e type Lyapunov function, can be found by integrating the above sector bounds.

It can be seen in Figure 3.2 and 3.3 that both the Lur'e and polynomial Lyapunov functions obtain ROA subsets which are close to the neighbouring unstable equilibrium point, a boundary of the actual ROA. There is a gap between the domain boundary and the region of attraction subset in Figure 3.3 as the plot is not the Lyapunov function but instead the bound.

The total computational time using polynomial Lyapunov functions (LF) was $3.0(s)$ *for the 1st iteration (*2^{nd} *order LF) and* $5.1(s)$ *for the* 2^{nd} *iteration (*4^{th} *order LF) using a* 2.66 *GHz Intel Core i7 Processor. The total computational time was* $3.4(s)$ *for the* 2^{nd} *order Lur'e-type LF. For this example the 2nd Order Lur'e Type Lyapunov function obtained a ROA subset with a similar area to the 4th order polynomial Lyapunov function, and so there is an advantage to using the Lur'e type as lower order Lyapunov functions can be constructed with less computational cost.*

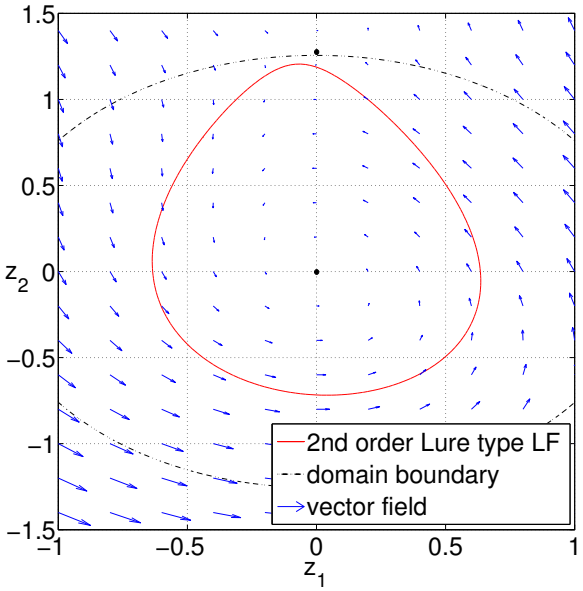


Figure 3.3: Region of Attraction subset using Lur'e type Lyapunov functions when $p = 0.8, \gamma = 0.4$. The stable equilibrium point is at $(0, 0)$ and an unstable equilibrium point is at $(0, 1.287)$. The vector field is described by Equation (3.16).

3.5 Further Examples

In this section we consider a further example, which shows a case when an uncertain system can be shown to be absolutely stable (for a global domain) when using generalised sectors but where the classical sector is inconclusive.

Example 3.15 *Consider the system:*

$$x = (x_1, x_2), \phi(y) = (\phi_1(y_1), \phi_2(y_2))$$

$$\dot{x} = f(x) = \begin{pmatrix} -5x_1 + 10x_2 + u_1 \\ -10x_1 - 2x_2 + u_2 \end{pmatrix}$$

$$y = h(x) = x$$

and

$$\begin{aligned} \alpha_1 &= 5x_1, \quad \beta_1 = 0, \quad A_1 = 1 + x_1^2, \quad B_1 = 1 \\ \alpha_2 &= 6x_2^2, \quad \beta_2 = 0, \quad A_2 = 1 + x_2^2, \quad B_2 = 1 \end{aligned} \tag{3.30}$$

which hold for $D_x = \mathbb{R}^n$. Theorem 3.7 can be implemented with SOSTOOLS to show that the system is absolutely stable where the returned 4^{th} order polynomial Lyapunov function is:

$$\begin{aligned} V(x) =& 0.64012x_1^2 + 0.07637x_1x_2 + 0.62894x_2^2 - 0.24557x_1^3 - 0.00158x_1^2x_2 - 0.35940x_1x_2^2 \\ & - 0.01098x_2^3 + 0.30470x_1^4 - 0.20933x_1^3x_2 + 0.69775x_1^2x_2^2 - 0.20371x_1x_2^3 + 0.34929x_2^4 \end{aligned}$$

This Lyapunov function is radially unbounded and so the system is globally asymptotically stable. This had a computational time of $2.0(s)$ using a 2.66 GHz Intel Core 2 Processor.

Using classical absolute stability then the tightest linear, global bounds are $\alpha_{cl}(x) = K_1x$ and $\beta_{cl}(x) = K_2x$ in (3.13), with

$$K_1 = \begin{pmatrix} 0 & 0 \\ 0 & -3 \end{pmatrix}, \quad K_2 = \begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix}$$

It can be seen that the bounds are the least conservative possible by noting that $\frac{6x_2^2}{1+x_2^2} = 3x_2 = 3$ at $x_2 = 1$ and $\frac{6x_2^2}{1+x_2^2} = -3x_2 = 3$ at $x_2 = -1$. One case within these bounds is:

$$\dot{x} = \begin{pmatrix} 0 & 10 \\ -10 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

which is unstable. Therefore, no conditions for global stability can be found for this case using classical absolute stability.

3.6 Conclusion

In this chapter the concept of absolute stability has been generalised by using SOS programming, an analysis tool for polynomial systems, rather than the traditional linear systems analysis tools. This has extended both absolute stability theory and the applicability of SOS programming. The chapter first showed how to use SOS programming to analyse

functions which are uncertain, time-varying or non-polynomial by using the generalised sector condition and by treating the functions as independent variables. This allows the problems to be converted to polynomial SOS problems. The developed techniques were then used together with generalised absolute stability theory to find stability conditions for nonlinear systems as well as region of attraction estimates. These techniques were applied to examples, which illustrated their usefulness.

Chapter 4

Perturbation Methods for Invariant Sum of Squares and Semidefinite Programming

4.1 Introduction

Dynamic networks are prevalent in many areas of science and technology; a number of techniques have been proposed to analyse these large scale systems, see, for example, [63, 88] and the references therein. To translate theoretical concepts for dynamical networks into useful tools, new computational techniques need to be developed. Computational methods for sum of squares, which was shown to be useful for polynomial systems in Chapter 2 and for nonlinear systems in Chapter 3, need to be extended for large-scale networked systems in order to capitalise on theoretical advances.

However, although SOS programming is very useful for analysing nonlinear systems, current Sum of Squares techniques do not scale well for large-scale systems. This scalability issue is the primary limitation stopping wider application of the technique in Science and Engineering. The factor limiting the scaling of the algorithm to larger systems is the size

of the resulting SDP, which is a combination of the size of the respective LMI and the number of decision variables. If the polynomial is 'sparse', i.e. many of the monomials have zero coefficients, then the size of the matrix and number of decision variables can be reduced significantly [43, 87]. Correlative sparsity, which uses the sparsity of the polynomial structure, can improve scalability [87], but the method is often conservative when the network is not a tree structure. Another form of sparsity uses Newton polytopes to limit the number of required monomials in the Gram matrix form [22, 53]. Structured Sum of Squares relaxations have been developed specifically for sensor networks using the sparsity of the LMI [64]. There are also a number of techniques to reduce the size of the LMI built into the preprocessing of current SOS computational tools [53].

Symmetric Sum of Squares and Semi-definite Programs can be computed with significant scalability improvement [27], but the technique cannot be used for SDP and SOS which are not invariant. Invariant or symmetric methods refer to SDP and SOS problems which are invariant under a set of transformations, where the transformations have a group structure. As well as scalability improvements, using symmetric methods also removes degeneracy, improves numerical conditions of SDPs and improves reliability as smaller SDPs are less prone to numerical error [27]. Symmetry in SPDs and SOS has been used in many applications (see [86] and references therein).

In this chapter we introduce perturbation methods for invariant Sum of Squares and Semi-Definite programs, which extends invariant methods to non-invariant polynomials and SDPs. This allows analysis of problems which would otherwise be computationally infeasible, as well as removing degeneracy, and improving conditioning and reliability. We apply the technique to testing non-negativity of polynomials or polynomial Lyapunov functions with a structure that is a perturbation from a symmetric network, in order to show the practical implementation of the technique. We show that the structured Sum of Squares scales significantly better than existing Sum of Squares techniques. For globally coupled networks we apply the technique for examples with over 100 variables. In comparison, existing Sum of Squares techniques are found to be impractical for networks

with 10-15 variables. In standard (primal) SDP, feasibility of solutions is tested, and then a solution to the SDP is found, which is minimisation of the cost function. In the proposed method, we obtain separate necessary and sufficient feasibility conditions and an upper and lower bound for the SDP, rather than an exact solution, as we are using perturbation methods. When using the method appropriately, the 'gap' can be shown to be small. The method is tested on perturbations of symmetric polynomials, but has the potential to improve scalability and other properties of large classes of SDPs. Due to the method's potential, we also discuss future directions in depth.

This chapter is organised as follows. In Section 4.2 we give background and in Section 4.3 we apply these existing methods to an example, which is used as a basis for examples later in the chapter. In Section 4.4 we develop the Perturbation Methods for invariant SOS and SDPs. In Section 4.5, we apply the methodology to networked polynomials. In Section 4.6, we apply the developed technique for networked polynomials to examples. In Section 4.7, we discuss future directions of research.

Notation: We use $\mathbf{S}^N$ to denote the vector space of real symmetric $N \times N$ matrices and $\mathbf{S}^N_+$ be the positive semi-definite subset of $\mathbf{S}^N$. The inner product (trace) $\langle X, Y \rangle = Tr(XY^T)$.

4.2 Background: Invariant Polynomials and SDPs

In this section we give background on techniques to exploit symmetry in semi-definite programming and sum of squares polynomials. We first give background on groups and symmetry, before showing applications in SDPs and SOS programming.

Groups and Representations [8]

Group theory is the natural mathematical setting for the study of symmetries and invariant transformations. To explain this area, we first start with the definition of a group.

Definition 4.1 *A group is a set G together with an operation $\cdot$ such that*

1. *If* $g_1, g_2 \in G$ *then* $g_1 \cdot g_2 \in G$

2. *If* $g_1, g_2, g_3 \in G$ *then* $g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$

3. *There exists* $e \in G$ *such that if* $g \in G$ *then* $e \cdot g = g \cdot e = g$

4. *If* $g_1 \in G$ *then then exists* $g_2 \in G$ *such that* $g_1 \cdot g_2 = g_2 \cdot g_1 = e$

Note: using typical notation, $g_1 g_2$ is used instead of $g_1 \cdot g_2$ and the group is named using the underlying set G.

We can also note that $|G|$ gives the order of the group i.e. the number of elements in the set.

One common example is the symmetry group S_N, which gives the set of all permutations of N elements $\{1, \ldots, N\}$ e.g. S_2, we have two group elements $(\{1, 2\}, \{2, 1\})$ while for S_3, we have a set of 6 permutations as we can swap 2 elements $(\{2, 1, 3\}, \{3, 2, 1\}, \{1, 3, 2\})$, we have the identity $\{1, 2, 3\}$ and we can rotate the 3 elements $(\{3, 1, 2\}, \{2, 3, 1\})$. Permutations are associative, have inverses and identities, and combine together to form other permutations, and so have a group structure.

Another common example of a group is the set of all real matrices which are invertible. This has the operation of matrix multiplication, and is called the general linear group over $\mathbb{R}$ $\big(GL(\mathbb{R})$ or $GL_n(\mathbb{R})\big)$. The first and fourth conditions hold as $A, B \in GL$ implies A, B are invertible and so $(AB)^{-1} = B^{-1}A^{-1}$ implies $AB \in GL$. The second and third conditions hold as matrix multiplication is associative and the identity matrix exists. This shows that $GL(\mathbb{R})$ has a group structure. Similarly, a general linear group of a vector space V $\big(GL(V)\big)$ is the group of all bijective linear transformations mapping $V \to V$.

We next define a homomorphism, which is a transformation that preserves a group structure.

Definition 4.2 *Let* G *and* H *be groups. A homomorphism is a function* $f : G \to H$ *such that if* $g_1, g_2 \in G$ *then*

$$f(g_1 g_2) = f(g_1) f(g_2) \tag{4.1}$$

The determinant of the group of invertible matrices $GL(\mathbb{R})$ is an example of a homomorphism as

$$\det(AB) = \det(A)\det(B) \tag{4.2}$$

for $A, B \in GL(\mathbb{R})$.

We next define a particular type of homomorphism, a representation, which is used to transform group operations and symmetries to a matrix framework. This gives a framework for symmetry in a linear algebra setting.

Definition 4.3 *A linear representation of a group G on a real vector space V is a homomorphism*

$$\sigma : G \to GL(V) \tag{4.3}$$

We use interchangeable notation for a linear representation $\sigma : g \to GL(V)$ as either $\sigma(g)$ or σ_g.

The representation is a set of transformations on vector space V, where all elements of the group are represented by transformations in the set and the set retains the same algebraic structure as the group.

Example 4.4 *We can use a representation to transform the symmetry group S_2 into the general linear group GL_2, where*

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A_s = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{4.4}$$

It can be seen that the algebraic structure is retained. For example, $\sigma(s)\sigma(s) = A_s A_s = I = \sigma(e) = \sigma(s \cdot s)$.

Symmetry in SDPs [27]

We next use the framework of group representation theory to study symmetry in semi-definite programs.

As mentioned in Chapter 2, the standard semi-definite program [21] can be given in the following way. Let $\mathcal{L}$ be an affine subset of $\mathbf{S}^N$, then the SDP problem is

$$F := \min_X \langle C, X \rangle \text{ such that } X \in \mathcal{L} \cap \mathbf{S}^N_+ \tag{4.5}$$

In the case of the Gram matrix Q for sum of squares polynomials, then $Q = X \in \mathcal{L}$ implies $p(x) = Z(x)^T Q Z(x)$ while $Q \in \mathbf{S}^N_+$ implies $Q \succeq 0$.

Definition 4.5 *[27] Given a finite group G, and associated linear representation $\sigma : G \to GL(\mathbf{S}^N)$:*

1. *A matrix X is invariant with respect to σ if $X = \sigma_g(X)$ for all $g \in G$.*

2. *A subspace $\mathcal{L}$ is invariant with respect to σ if $X \in \mathcal{L}$ implies $\sigma_g(X) \in \mathcal{L}$ for all $g \in G$.*

3. *A cost function $\langle C, X \rangle$ is invariant with respect to σ if $\langle C, X \rangle = \langle C, \sigma_g(X) \rangle$ for all $g \in G$.*

4. *An SDP of the form (4.5) is called invariant with respect to σ, if $\mathcal{L} \cap \mathbf{S}^N_+$ and the cost function $\langle C, X \rangle$ are invariant with respect to σ.*

Most of the transformations σ which we are interested in are induced by the linear representation $\rho : G \to GL(\mathbb{R}^N)$, such that

$$\sigma_g(X) = \rho_g^T X \rho_g \text{ for all } g \in G \tag{4.6}$$

This form ensures that $\sigma_g(X) = \rho_g^T X \rho_g \succeq 0$ if $X \succeq 0$.

Example 4.6 *For the group S_2 with a representation $\sigma_g(X) = \rho_g^T X \rho_g$ and*

$$\rho_e = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \rho_s = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.7}$$

then the matrix $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ *is invariant wrt σ. The subspace described by* $\begin{bmatrix} x_1 & 1 \\ 1 & x_2 \end{bmatrix} \succeq 0$ *where x_1, x_2 are scalar variables is invariant wrt σ. The cost function with $C = I$ is invariant wrt σ. The SDP*

$$\min x_1 + x_2 \text{ such that } \begin{bmatrix} x_1 & 1 \\ 1 & x_2 \end{bmatrix} \succeq 0 \tag{4.8}$$

is invariant wrt σ.

We next provide a definition followed by a theorem which is useful in exploiting invariance to simplify an SDP.

Definition 4.7 *[27] A fixed point subspace of $\mathbf{S}^N$ is the subspace $\mathcal{F}$ of all invariant matrices*

$$\mathcal{F} = \{X \in \mathbf{S}^N | X = \sigma_g(X)\} \tag{4.9}$$

and the associated semi-definite program

$$F_\sigma := \min_X \langle C, X \rangle \text{ such that } X \in \mathcal{F} \cap \mathcal{L} \cap \mathbf{S}_+^N \tag{4.10}$$

is obtained by restricting the feasible set of (4.5) *to the fixed-point subspace $\mathcal{F}$.*

Theorem 4.8 *[27] Given a linear representation $\sigma : G \to GL(\mathbf{S}^N)$ of a finite group G, consider an SDP which is invariant with respect to σ. If there exists $X \in \mathcal{L} \cap \mathbf{S}_+^N$ then there exists $X \in \mathcal{F} \cap \mathcal{L} \cap \mathbf{S}_+^N$. Also, the optimal values F and F_σ of the original SDP* (4.5) *and the restricted SDP* (4.9) *are equal.*

Example 4.9 *Consider the SDP and σ in Example 4.6. By taking the fixed point subspace*

then $x_1 = x_2 := x$ *and so from Theorem 4.8, we have a simplified SDP*

$$\min 2x \text{ such that } \begin{bmatrix} x & 1 \\ 1 & x \end{bmatrix} \succeq 0 \tag{4.11}$$

with the same solution as the SDP in Example 4.6.

We can further extend Theorem 4.8 if we assume that σ_g is of the form (4.6) and $\rho_g^T = \rho_g^{-1}$, which allows a block diagonalisation $\bar{X}$ using a generated transformation.

We first give the applicable equations, followed by a conceptual explanation. The block diagonalising transformation has the form

$$X_{bl} = T^T \bar{X} T \tag{4.12}$$

where X_{bl} is block diagonal.

To find the transformation T, we need to first introduce irreducible representation and invariance subspaces for vectors, as opposed to matrices. Now a G-invariant subspace in terms of vectors, rather than matrices, is defined by $\rho_g W \subset W \subset \mathbb{R}^n$. Irreducible representation are representations restricted to invariant subspaces, where the subspaces cannot be further decomposed. We can generate T using the canonical transformation [75] for each irreducible representation

$$\mathcal{P} = \sum_{g \in G} \chi_g \rho_g \tag{4.13}$$

where χ_g is the character (trace) of the irreducible representation, . The orthonormal vectors spanning the image space of each $\mathcal{P}$ make up the columns of T. The characters can be found from the irreducible representations or from a character table. If an irreducible representation is repeated, then there are non-unique methods to further decompose the space, which further block diagonalises Q [75].

The block diagonalisation of $\bar{X}$ can be conceptually explained by using eigenvector/value

decomposition as an analogy, although the method of implementation differs. In finding eigenvectors/values $Av_i = \lambda_i v_i$, we find a linear subspace spanned by a vector v_i which is invariant to transformation A. By transforming to a set of coordinates based on the invariant spaces for eigenvalues/eigenvectors, we can (block) diagonalise the matrix. Analogously for group representations, we find invariant subspaces with respect to ρ_g. By finding a basis to these subspaces using the above methodology, we can find transformations which block diagonalise X. The technique differs from eigenvalue/eigenvector analysis as we can use the same transformation to block diagonalise all X in the subspace $\mathcal{L} \cap \mathbf{S}^N$ and hence is applicable to SDPs. The technique relies on Schur's Lemma, further details of which can be found in [8] and [75].

We first show the block diagonalisation of a matrix.

Example 4.10 *Consider the block matrix*

$$X = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \tag{4.14}$$

which is invariant wrt the representation $\sigma_g(X) = \rho_g^T X \rho_g$ *with*

$$\rho_e = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \rho_s = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \tag{4.15}$$

We can use (4.13) *and the character table of group* S_2 *[8] to generate*

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \tag{4.16}$$

which gives

$$X_{bl} = T^T X T = \begin{bmatrix} A+B & 0 \\ 0 & A-B \end{bmatrix} \tag{4.17}$$

We next give the block diagonalisation for SDPs.

Theorem 4.11 *[27] Let G be a finite group with h real irreducible representations. Let σ be a linear representation of G on $\mathbf{S}^N$ induced as in (4.6) by an orthogonal linear representation ρ on $\mathbb{R}^N$. Given an SDP of the form (4.5) which is invariant with respect to σ then there exists an explicitly computable co-ordinate transformation under which all invariant matrices are block diagonal. As a consequence, the SDP can be simplified to*

$$\min_{X\in\mathcal{L}^N, X_i\in\mathcal{S}_+^{m_i}} \sum_{i=1}^{h} m_i\langle C_i, X_i\rangle \tag{4.18}$$

where m_i are the multiplicities in ρ and n_i are the dimensions of the irreducible representations.

Example 4.12 *Following on from Example 4.9, consider the SDP and ρ in Example 4.6. Using $T = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, we find the simplified SDP*

$$\min 2x \text{ such that } \begin{bmatrix} x+1 & 0 \\ 0 & x-1 \end{bmatrix} \succeq 0 \tag{4.19}$$

which further simplifies to

$$\min 2x \text{ such that } x \geq 1 \tag{4.20}$$

Invariant Polynomials and SOS

We next show applications of SDP invariance to polynomials, which is the case study used throughout the chapter.

Definition 4.13 *A polynomial $p(x)$ is invariant with respect to linear representation $\theta : G \to GL(\mathbb{R}^N)$ if $p(x) = p(\theta_g x)$.*

Invariant polynomials can be used to determine the representations for Theorem 4.8 and 4.11 by noting that the linear representation θ can induce a linear representation ρ, where

$Z(\theta_g x) = \rho_g Z(x)$ and so for an invariant polynomial

$$p(x) = p(\theta_g x) = Z(\theta_g x)^T Q Z(\theta_g x) = Z(x)^T \rho_g^T Q \rho_g Z(x) \tag{4.21}$$

which implies $\mathcal{L} \cap \mathbf{S}_+^N$ is invariant with respect to σ. Using the invariance to find a block diagonalising transformation, we have

$$\begin{aligned} p(x) &= Z(x)^T \bar{Q} Z(x) \\ &= Z(x)^T T \left(T^T \bar{Q} T \right) T^T Z(x) \\ &= Y(x)^T Q_D Y(x) \end{aligned} \tag{4.22}$$

where $\bar{Q}$ is the fixed point of Q, $Q_D = T^T \bar{Q} T$ is block diagonal and $Y(x) = T^T Z(x)$.

Example 4.14 *Taking the case of sign symmetry, if $p(x) = p(-x)$ then we have two transformations in the group, the identity and $\theta_s x = -x$. We can write the Gram matrix as*

$$p(x) = Z(x)^T Q Z(x) = \begin{pmatrix} Z_e^T & Z_o^T \end{pmatrix} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} Z_e \\ Z_o \end{pmatrix} \tag{4.23}$$

where Z_e are the even monomials and Z_o represent the odd monomials. Now we have $Z(\theta_s x) = Z(-x) = (Z_e, -Z_o)$ and so

$$p(x) = p(-x) = Z(-x)^T Q Z(-x) = Z(x)^T \rho_s^T Q_s \rho_s Z(x) \tag{4.24}$$

$$\rho_s = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}, \rho_s^T Q_s \rho_s = \begin{pmatrix} A & -B \\ -B^T & C \end{pmatrix} \tag{4.25}$$

Taking the fixed point $Q_s = \rho_s^T Q_s \rho_s$ results in $B = 0$ and the simplified Gram Matrix

$$\bar{Q} = \begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix} \tag{4.26}$$

which is block diagonalised and has fewer parameters as $B = 0$. Intuitively, this makes

sense, as for $Q \succeq 0$ we require $A, C \succeq 0$ and, noting that even monomials multiplied by odd monomials are odd, we can ignore the off diagonal blocks as their contribution to $p(x)$ sums to zero [53].

In this case $T = I$ in the block diagonalisation $T^T QT$. However, if we had not partitioned $Z(x)$ into even and odd monomials, then $Q_D = T^T QT$ and $Y(x) = T^T Z(x)$ would have involved swapping rows and columns to partition even and odd monomials in $Z(x)$.

4.3 A Networked Symmetric Polynomial using a Representation of S_3

In this section we apply methods for SOS programming of symmetric polynomials to the case of networked polynomials. We define networked polynomials to be of the form

$$p(x_1, \ldots, x_N) = \sum_{i=1}^{N} p_i(x_i) + \sum_{i=1}^{N} \sum_{k=1}^{N} b_{ik} p_c(x_i, x_k) \tag{4.27}$$

where $x \in \mathbb{R}^N$, the term $b_{ij} p_c(x_i, x_j)$ represents the links of the network and $p_i(x_i)$ represents the nodes.

We look at the case of a three node symmetric network, noting that the methodology can be generalised to networks of any size. This example applies existing theory, but differs from standard applications by manipulating the Gram matrix form, rather than the polynomials directly. Also, the reduced form is more general than standard examples, i.e. for a general 4^{th} order networked polynomial rather than a specific 4^{th} order polynomial. This example is used as a foundation later in the chapter for larger networks which are not necessarily symmetric or invariant.

Consider the 4^{th} order polynomial

$$p(x_1, x_2, x_3) = g_c(x_1, x_2) + g_c(x_1, x_3) + g_c(x_2, x_3) \tag{4.28}$$

In this case, we can permute the 3 variables (x_1, x_2, x_3), with the set of permutations making up the S_3 group.

This can be written in Gram matrix form with

$$Z(x)^T = (Z_1^T, Z_2^T, Z_3^T, Z_{12}^T, Z_{13}^T, Z_{23}^T) \tag{4.29}$$

where Z_i represents monomials of x_i only, Z_{ij} represents monomials with both x_i and x_j. Using Newton polytopes [22, 53], we can show that these monomials are sufficent for the Gram matrix representation of these 4^{th} order networked polynomials. For this case, the monomials are of the form

$$Z_i = \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix}, \; Z_{ij} = [x_i x_j] \tag{4.30}$$

The Gram Matrix can be written as

$$\bar{Q} = \left[\begin{array}{ccc:ccc} A_1 & B_1 & B_2 & C_1 & C_2 & J_1 \\ B_1^T & A_2 & B_3 & C_3 & J_2 & C_4 \\ B_2^T & B_3^T & A_3 & J_3 & C_5 & C_6 \\ \hdashline C_1^T & C_3^T & J_3^T & E_1 & F_1 & F_2 \\ C_2^T & J_2^T & C_5^T & F_1^T & E_2 & F_3 \\ J_1^T & C_4^T & C_6^T & F_2^T & F_3^T & E_1 \end{array}\right] \tag{4.31}$$

where the blocks contain appropriate constants and free parameters such that $p(x) = Z(x)^T Q Z(x)$. In this, the matrices have the form.

We use the transformation $\sigma_g(Q) = \rho_g^T Q \rho_g$, which permutes the appropriate rows and

columns of blocks. For example, to swap $x_1 \leftrightarrow x_2$, we use

$$\rho_{g1} = \left[\begin{array}{ccc:ccc} 0 & I_n & 0 & 0 & 0 & 0 \\ I_n & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & I_c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_c \\ 0 & 0 & 0 & 0 & I_c & 0 \end{array}\right] \tag{4.32}$$

We next take the fixed point from the set of transformations, which obtains

$$\bar{Q}_f = \left[\begin{array}{ccc:ccc} A & B & B & C & C & J \\ B & A & B & C & J & C \\ B & B & A & J & C & C \\ \hdashline C^T & C^T & J^T & E & F & F \\ C^T & J^T & C^T & F & E & F \\ J^T & C^T & C^T & F & F & E \end{array}\right] \tag{4.33}$$

where $B = B^T$ and $F = F^T$.

This method reduces the number of decision variables in the equivalent SDP, but to reduce the size of the system we need to find a transformation to block diagonalise $\bar{Q}$. We complete this using the symmetry adapted transformation T, where $T^T T = T T^T = I$ and so

$$p(x) = (Z(x)^T T)(T^T \bar{Q}_f T)(T^T Z(x)) = Y(x)^T (T^T \bar{Q}_f T) Y(x) \tag{4.34}$$

where $Y(x) = T^T Z(x)$ and $(T^T \bar{Q} T)$ is block diagonal. Using

$$T = \left[\begin{array}{cc:cc:cc} \alpha I_n & 0 & \gamma I_n & 0 & \beta I_n & 0 \\ -\alpha I_n & 0 & \gamma I_n & 0 & \beta I_n & 0 \\ 0 & 0 & -2\gamma I_n & 0 & \beta I_n & 0 \\ \hdashline 0 & 0 & 0 & -2\gamma I_c & 0 & \beta I_c \\ 0 & -\alpha I_c & 0 & \gamma I_c & 0 & \beta I_c \\ 0 & \alpha I_c & 0 & \gamma I_c & 0 & \beta I_c \end{array}\right] \tag{4.35}$$

where $\beta = \frac{1}{\sqrt{3}}$, $\alpha = \frac{1}{\sqrt{2}}$, $\gamma = \frac{1}{\sqrt{6}}$, and I_n, I_c are the identity of appropriate dimension, we obtain

$$T^T \bar{Q}_f T = \left[\begin{array}{cc:cc:cc} A-B & J-C & 0 & 0 & 0 & 0 \\ J^T - C^T & E-F & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & A-B & J-C & 0 & 0 \\ 0 & 0 & J^T - C^T & E-F & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & A+2B & 2C+J \\ 0 & 0 & 0 & 0 & 2C^T + J^T & E+2F \end{array}\right] \tag{4.36}$$

This block diagonalised form gives a smaller SDP, with repeated blocks reducing the problem further as only one of the repeats needs be calculated.

Example 4.15 *For*

$$g_{ij}(x_i, x_j) = \frac{1}{2}(x_i^2 + x_i^4 + x_j^2 + x_j^4) - x_i^2 x_j^2 \tag{4.37}$$

in (4.28) *then we have*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0 & \lambda_2 \\ \lambda_2 & \lambda_1 \end{bmatrix} \tag{4.38}$$

$$C = \begin{bmatrix} -\lambda_2 \\ 0 \end{bmatrix}, \; J = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; E = [-2\lambda_1 - 1], \; F = [0], \tag{4.39}$$

where the decision variables are required due to non-unique representations from $(x_i^2)(x_j^2) = (x_i x_j)^2$ *and* $(x_i^2)x_j = x_i(x_i x_j)$*, noting that other decision variables cancel due to symmetry. Thus we have*

$$\begin{bmatrix} A - B & J - C \\ J^T - C^T & E - F \end{bmatrix} = \begin{bmatrix} 1 & -\lambda_2 & \lambda_2 \\ -\lambda_2 & 1 - \lambda_1 & 0 \\ \lambda_2 & 0 & -2\lambda_1 - 1 \end{bmatrix} \tag{4.40}$$

$$\begin{bmatrix} A + 2B & 2C + J \\ 2C^T + J^T & E + 2F \end{bmatrix} = \begin{bmatrix} 1 & 2\lambda_2 & -2\lambda_2 \\ 2\lambda_2 & 1 + 2\lambda_1 & 0 \\ -2\lambda_2 & 0 & -2\lambda_1 - 1 \end{bmatrix} \tag{4.41}$$

which is positive semi-definite for $\lambda_1 = -\frac{1}{2}$*,* $\lambda_2 = 0$*.*

4.4 Perturbation Methods for Invariant SDP and Polynomials

In order to study sum of squares and SDPs using computationally tractable methods, we find conditions in terms of invariant matrices and polynomials. In this section we develop a perturbation method for invariant Sum of Squares and Semidefinite programs. Given a non-invariant SDP, we use a representation to generate an invariant SDP, which gives necessary conditions for feasibility and a lower bound for the original SDP. We then perturb the generated invariant SDP, which gives sufficient conditions for feasibility and an upper bound of the original SDP. The invariant SDP and its perturbation have reduced computational complexity over the original SDP, often significantly so. Any representation may be used for the feasibility of any SDP, as the perturbation is based on

convexity rather than a 'small' variable (e.g. as used in linearisation). However, using an appropriate representation is required to reduce the 'gap' between necessary and sufficient conditions for feasibility and the upper and lower bounds of the original SDP. We use Sum of Squares as an example of application.

4.4.1 Generating Invariant SDPs and Polynomials: Necessary Conditions and Lower Bounds

We first present a method of taking an SDP or polynomial and generating a related invariant SDP or polynomial. This is achieved by using an SDP version of the averaging method used in group representation theory [8]. The generated SDP or polynomial replaces the fixed point subspace in methods for invariant SDPs. We first use the generated SDP to find necessary conditions for feasibility and a lower bound of the original SDP.

Lemma 4.16 *Given a linear representation $\sigma : G \to GL(\mathbf{S}^N)$ of a finite group G, where $X \succeq 0$ implies $\sigma_g(X) \succeq 0$, then*

$$\bar{X} = \frac{1}{|G|} \sum_{g \in G} \sigma_g(X) \succeq 0 \tag{4.42}$$

and $\bar{X}$ and $\langle C, \bar{X} \rangle$ are invariant with respect to σ.

Proof. If $X \succeq 0$ then $\sigma_g(X) \succeq 0$ for all $g \in G$ and so $\bar{X} \succeq 0$ by convexity. Now for all $h \in G$ then

$$\sigma_h \bar{X} = \frac{1}{|G|} \sum_{g \in G} \sigma_h \sigma_g \bar{X} \tag{4.43}$$

For all $g, h \in G$ then $\sigma(h)\sigma(g) = \sigma(hg) = \sigma(g')$ for some $g' \in G$ as $hg \in G$. Also, h has an inverse, and so $g' = hg$ is distinct for all $g \in G$. Thus

$$\sigma_h \bar{X} = \frac{1}{|G|} \sum_{g' \in G} \sigma_{g'} X = \bar{X} \tag{4.44}$$

As $\bar{X}$ is invariant with respect to σ then $\langle C, \bar{X}\rangle = \langle C, \sigma_g \bar{X}\rangle$ and so $\langle C, \bar{X}\rangle$ is invariant with respect to σ. ■

Restating the SDP definition, let $\mathcal{L}$ be an affine subset of $\mathbf{S}^N$

$$F := \min_X \langle C, X\rangle \text{ such that } X \in \mathcal{L} \cap \mathbf{S}^N_+ \tag{4.45}$$

By averaging as per (4.42) using a linear representation $\sigma : G \to GL(\mathbf{S}^N)$ such that $\mathcal{U} = \{\bar{X} \in \mathbf{S}^N | \bar{X} = \frac{1}{|G|}\sum_{g\in G} \sigma_g(X), X \in \mathcal{L}\}$, we obtain the corresponding problem

$$\bar{F}_l := \min_X \langle C, X\rangle \text{ such that } X \in \mathcal{U} \cap \mathbf{S}^N_+ \tag{4.46}$$

We can note that by construction, $\mathcal{U}$ is a fixed space, which means that $X = \sigma_g(X)$ for all $X \in \mathcal{U}$.

We next use the invariant SDP to give a lower bound of the original SDP.

Theorem 4.17 *Given a linear representation $\sigma : G \to GL(\mathbf{S}^N)$ of a finite group G where $X \succeq 0$ implies $\sigma_g(X) \succeq 0$ for all $g \in G$, and $\langle C, X\rangle \geq \langle C, \sigma_g(X)\rangle$ for all $g \in G$ then $F \geq \bar{F}_l$.*

Proof. As $X \succeq 0$ implies $\sigma_g(X) \succeq 0$, then by definition $X \in \mathcal{L} \cap \mathbf{S}^N_+$ implies $\bar{X} \in \mathcal{U} \cap \mathbf{S}^N_+$. Now $\langle C, X\rangle \geq \langle C, \bar{X}\rangle$ for all $X \in \mathcal{L} \cap \mathbf{S}^N_+$ and so $F \geq \bar{F}_l$. ■

We also give the equivalent SOS result.

Theorem 4.18 *If $p(x)$ is a polynomial SOS then for any linear representation $\theta : G \to GL(\mathbb{R}^N)$ of a group G we have*

$$\bar{p}(x) = \frac{1}{|G|}\sum_{g\in G} p(\theta(g)x) \text{ is SOS and invariant wrt } \theta \tag{4.47}$$

Proof. The proof is as per the proof of Lemma 4.16. ■

We next give a simple illustrative example.

Example 4.19 *The polynomial* $p(x) = x^2 + x^3 + x^4 = x^2(1 + \frac{1}{2}x)^2 + \frac{3}{4}x^4$ *is SOS. Taking a sign transformation, we also know that* $p(-x) = x^2 - x^3 + x^4$ *is SOS. Taking the average, we have* $\bar{p}(x) = \frac{1}{2}p(x) + \frac{1}{2}p(-x) = x^2 + x^4$ *is SOS. As* $\bar{p}(x) = x^2 + x^4 = \bar{p}(-x)$ *then* $\bar{p}(x)$ *is invariant with respect to the sign symmetry.*

Note: The group is $\{1, s\}$ *where* $s^2 = 1$*, with representation* $\theta(1) = 1$ *and* $\theta(s) = -1$*.*

4.4.2 Unstructured Perturbations of Invariant Polynomials and SDPs: Sufficient Conditions

We next use the subspace generated by the averaging method to find sufficient conditions for feasibility of an SDP. Using SOS as a case study, if we find a set of transformations θ such that $\bar{p}$ and p are not too dissimilar then the 'gap' between necessary and sufficient conditions is small. We first use Theorem 4.18 to determine necessary conditions by determining whether $\bar{p}$ is SOS. We next determine whether perturbation of $\bar{p}$ is a Sum of Squares, which gives sufficient conditions. Analysing $\bar{p}$ can give a significant computational advantage as invariant polynomials may be analysed by exploiting symmetry.

We first attempt to determine sufficient conditions for sum of squares using unstructured perturbations from the invariant polynomials and SDPs. For the case where we cannot show equivalence, then we also want to use the structure of $\bar{p}(x)$ for sufficient conditions, where we use the efficient methodology to determine whether $\bar{p}(x)$ is a Sum of Squares in order to show that $p(x)$ is a sum of squares.

A first naive attempt may test whether $\bar{p}(x)$ is sum of squares, and assume that if $p(x)$ and $\bar{p}(x)$ are close, then $p(x)$ is also a sum of squares. However, this is not necessarily the case.

Example 4.20 *For the case* $p(x) = 2bx + x^4$*, if we use a sign transformation then* $\bar{p}(x) = x^4$ *is close to* $p(x)$ *for small* b*. However, in this case,* $\bar{p}(x)$ *is SOS while* $p(x)$ *is not SOS for any* $b \neq 0$*.*

A more reasonable attempt to study SOS of $p(x)$ using $\bar{p}(x)$ is to show $\bar{p}(x)$ is SOS with a small degree of robustness. Now if $\bar{p}(x)$ is sum of squares then there is a $\bar{Q} \succeq 0$ where $\bar{p}(x) = Z(x)^T \bar{Q} Z(x)$. If $\bar{Q}$ can be found such that all eigenvalues $\lambda_i > 0$ for $\bar{Q}$, then $\bar{Q}$ may have a small, arbitrary perturbation and still remain positive semi-definite [28]. Thus requiring positive eigenvalues of $\bar{Q}$ rather than non-negative eigenvalues, gives a small degree of robustness.

Recall that for the affine set $\mathcal{L} \subset \mathcal{S}^N$ then $d(A, \mathcal{L}) = \inf_{B \in \mathcal{L}} \|A - B\|$, and note that in general $\mathcal{U} \neq \mathcal{L}$.

Corollary 4.21 *If $d(\bar{Q}, \mathcal{L}) \leq k$ for all $\bar{Q} \in \mathcal{U}$ and there exists $\bar{Q} \in \mathcal{U}$ such that $\bar{Q} \succeq kI$ then $Q \succeq 0$ for some $Q \in \mathcal{L}$.*

Proof. $d(\bar{Q}, \mathcal{L}) \leq k$ implies $-kI \preceq Q - \bar{Q} \preceq kI$ for some $Q \in \mathcal{L}$ and so there exists $Q = \bar{Q} + (Q - \bar{Q}) \succeq 0$.

■

Example 4.22 *For the case $p(x) = a + 2bx + x^4$, if we use a sign transformation then we have $\bar{p}(x) = a + x^4$ is close to $p(x)$ for small b. In Gram Matrix form, we have*

$$Z(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}, \bar{Q} = \begin{bmatrix} a & 0 & -q \\ 0 & 2q & 0 \\ -q & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} a & b & -q \\ b & 2q & 0 \\ -q & 0 & 1 \end{bmatrix} \tag{4.48}$$

where q is a non-unique parameter as $1 \cdot x^2 = x \cdot x$. For the case in Example 4.19 where $a = 0$ then $q = 0$ is the only solution and so $\bar{Q}$ has eigenvalues $\lambda_1, \lambda_2 = 0, \lambda_3 = 1$. However, for $a > 0$ then $q > 0$ can be chosen such that all eigenvalues of $\bar{Q}$ are positive. Thus arbitrary perturbations of $\bar{Q}$ are positive semi-definite, and so there exists $Q \succeq 0$ for small b. Therefore, we know that $p(x)$ is SOS for small b.

Conditions requiring positive eigenvalues are not useful for many Sum of Squares programming problems, however this is not necessarily the case for invariant polynomials. For standard SOS programming problems, the returned Gram matrix often has (multi-

ple) zero eigenvalues [53], and so cannot be used to show robustness. However, exploiting symmetry in polynomials allows degeneracy removal [27], and so intuitively we would expect the eigenvalue conditions to be of more practical use as long as we have chosen an appropriate set of transformations θ.

4.4.3 Structured Perturbations of Invariant Polynomials and SDPs: Sufficient Conditions and Upper Bounds

To further improve methods of perturbing invariant SDPs, we use the structured perturbations to the invariant SDP. For this case we need to find a transformation such that $\Phi(\bar{Q}) \in \mathcal{L}$ for all $\bar{Q} \in \mathcal{U}$, with corresponding space $\mathcal{B} = \{X \in \mathbf{S}^N | X = \Phi(\bar{Q}), \bar{Q} \in \mathcal{U} \cap \mathbf{S}^N\}$. As $\mathcal{B} \subset \mathcal{L}$ then we obtain sufficient conditions for feasibility of the original SDP. We also obtain the corresponding SDP problem

$$\bar{F}_u := \min_X \langle C, X \rangle \text{ such that } X \in \mathcal{B} \cap \mathbf{S}_+^N \tag{4.49}$$

Theorem 4.23 *Given a linear representation $\sigma : G \to GL(\mathbf{S}^N)$ of a finite group G then $F \leq \bar{F}_u$.*

Proof. As $\mathcal{B} \subset \mathcal{L}$ then the result follows trivially. ■

We can note that requiring $\Phi(\bar{Q}) \in \mathcal{L}$ for all $\bar{Q} \in \mathcal{U}$ is equivalent to requiring $\langle A_i, \Phi(\bar{Q}) \rangle = b_i$ for all i and all $\bar{Q} \in \mathcal{U}$ (See SDP form (2.26), instead of (4.5)).

One general choice of transformation $\Phi(\cdot)$ is the Schur product (or Hadamard transformation) to make an affine transformation to each element. For this transformation we have

$$Q_F = \Phi(\bar{Q}) = R + P \circ \bar{Q} \tag{4.50}$$

such that $\langle A_i, Q_F \rangle = b_i$ for all i and all $\bar{Q} \in \mathcal{U}$. For simplicity, in this chapter we use (4.50) along with the extra condition

$$\langle A_i, P \circ \bar{Q} \rangle = b_i \text{ when } \langle A_i, \bar{Q} \rangle \neq 0 \tag{4.51}$$

for all i. Any free parameters in Q_F can be chosen arbitrarily, although we typically set free parameters to 1 or 0.

Example 4.24 *Consider* $p(x) = a + 2bx + x^4 + 2cx^5 + x^6$. *For this case, the Gram matrix is*

$$Z(x) = \begin{bmatrix} 1 \\ x^2 \\ x \\ x^3 \end{bmatrix}, Q = \begin{bmatrix} a & -q_2 & b & -q_1 \\ -q_2 & 1+2q_3 & q_1 & c \\ b & q_1 & 2q_2 & -q_3 \\ -q_1 & c & -q_3 & 1 \end{bmatrix} \tag{4.52}$$

where q_1, q_2, q_3 *are free parameters. For simplicity, we have chosen* $Z(x)$ *to be grouped as even and odd polynomials. Using a sign transformation, we have the invariant polynomial* $\bar{p}(x)$, *with block diagonalised Gram matrix*

$$\bar{Q} = \left[\begin{array}{cc:cc} a & -q_2 & 0 & 0 \\ -q_2 & 1+2q_3 & 0 & 0 \\ \hdashline 0 & 0 & 2q_2 & -q_3 \\ 0 & 0 & -q_3 & 1 \end{array}\right], Z(x) = \left[\begin{array}{c} 1 \\ x^2 \\ \hdashline x \\ x^3 \end{array}\right] \tag{4.53}$$

Now if $p(x)$ *is SOS then we can find* $\bar{Q} \succeq 0$, *and so* (4.53) *serves as the necessary conditions.*

We can find Q_F required for sufficient conditions by setting $P = \text{ones}(4)$ and

$$R = \left[\begin{array}{cc:cc} 0 & 0 & b & 0 \\ 0 & 0 & 0 & c \\ \hdashline b & 0 & 0 & 0 \\ 0 & c & 0 & 0 \end{array}\right], \tag{4.54}$$

in (4.50) *to obtain*

$$Q_F = \left[\begin{array}{cc:cc} a & -q_2 & b & 0 \\ -q_2 & 1+2q_3 & 0 & c \\ \hdashline b & 0 & 2q_2 & -q_3 \\ 0 & c & -q_3 & 1 \end{array}\right], \tag{4.55}$$

with a reduced number of decision variables. Now $Z(x)^T Q_F Z(x) = p(x)$ and so if there exists $Q_F \succeq 0$ then $p(x)$ is SOS, and so (4.55) *serves as the sufficient conditions.*

4.4.4 Block Diagonalisation

We next wish to exploit Schur's Lemma to block diagonalise the generated SDPs $\bar{F}_u$ and $\bar{F}_l$. The necessary conditions and lower bound $\bar{F}_l$ are block diagonalisable as they are invariant, but for the sufficient conditions and upper bound $\bar{F}_u$ this is not necessarily true. However, for appropriate transformation, then $T^T Q_F T$ is 'block-diagonally dominant', which means that the 'off block diagonal' terms are small for a small perturbation rather than zero. We next use the 'block-diagonally dominant' property to reduce the size of the SDPs.

Definition 4.25 *Symmetric matrix A is block diagonally dominant if*

$$\|A_{ii}^{-1}\|^{-1} \geq \sum_{i=1, j\neq i} \|A_{ij}\| \tag{4.56}$$

for some matrix norm $\|\cdot\|$, where A_{ij} is a block of A.

To improve the block diagonalisation of $T^T Q_F T$ and reduce off diagonal terms, we can transform T and Q_F, testing

$$H(T)^T K(Q_F) H(T)^T \succeq 0 \tag{4.57}$$

where K and H are bijective transformations.

The simplest case of perturbing T is to use diagonal W such that $H_T(T) = WT$, where W is the weighting matrix. This gives the polynomial

$$p(x) = \left[Z(x)^T (W^{-1})^T T\right] \left[T^T W^T Q_F W T\right] \left[T^T W^{-1} Z(x)\right] \tag{4.58}$$

where $Y(x) = T^T W^{-1} Z(x)$ and we require

$$Q_{bl} = T^T W^T Q_F W T \succeq 0 \tag{4.59}$$

where Q_{bl} is assumed to be a block diagonally dominant matrix.

One method to perturb K_f is through the Hadamard product $K(Q_F) = R \circ Q_f$, for the case where there exists $\Gamma \succeq 0$ such that $\Gamma \circ R \circ Q_F = Q_F$. For this transformation $K(Q_f) \succeq 0$ implies $Q_F \succeq 0$.

Once we have a block diagonally dominant matrix, we next need to convert it to a form useful for SDPs.

Lemma 4.26 *For symmetric matrix A, if*

$$\sum_{i=1, j \neq i} \|A_{ij}\| \leq k_i \tag{4.60}$$

for some matrix norm $\|\cdot\|$ and

$$A_{ii} \succeq k_i I \tag{4.61}$$

then $A \succeq 0$.

Proof.

For $k_i > 0$, if $A_{ii} \succeq k_i I$ then $A_{ii}^{-1} \preceq k_i^{-1} I$. Thus $\|A_{ii}^{-1}\| \leq k_i^{-1}$ and $\|A_{ii}^{-1}\|^{-1} \geq k_i$. The remainder of the proof follows standard results [36]. The case of $k_i = 0$ is trivial. ■

We next use the block diagonally dominant result in an example.

Example 4.27 *Following on from Example 4.24, we have*

$$Q_F = \left[\begin{array}{cc:cc} a & -q_2 & b & 0 \\ -q_2 & 1+2q_3 & 0 & c \\ \hdashline b & 0 & 2q_2 & -q_3 \\ 0 & c & -q_3 & 1 \end{array}\right], \tag{4.62}$$

In this case we set $T = I$ as the even and odd monomials are already partitioned. If they were not, $T^T Q_F T$ would swap rows and columns of Q_F. We can test the matrices in two ways.

Firstly, we can assume that the block diagonals are block diagonally dominant if b, c are small and so test

$$\left[\begin{array}{cc} a & -q_2 \\ -q_2 & 1+2q_3 \end{array}\right], \left[\begin{array}{cc} 2q_2 & -q_3 \\ -q_3 & 1 \end{array}\right] \succeq 0 \tag{4.63}$$

We can then confirm that $Q_F \succeq 0$ once we have found particular choices for q_2, q_3. We can also use the block-diagonally dominant property of Q_F for small b, c. Using Lemma 4.26 and setting $k = \max(|b|, |c|)$, we test

$$\left[\begin{array}{cc} a-k & -q_2 \\ -q_2 & 1+2q_3-k \end{array}\right], \left[\begin{array}{cc} 2q_2-k & -q_3 \\ -q_3 & 1-k \end{array}\right] \succeq 0 \tag{4.64}$$

We also need to be able to deal with variables in the off-diagonal blocks, for which we use the following lemma.

Lemma 4.28 *For symmetric, parameterised matrix $A = A^{(0)} + \sum_m q_m A^{(m)}$, constants*

$k_i^{(m)}$, *and some matrix norm* $\|\cdot\|$, *if*

$$\sum_{i=1,j\neq i} \|A_{ij}^{(m)}\| \leq k_i^{(m)} \tag{4.65}$$

$$A_{ii} \succeq \left(k_i^{(0)} + \sum_m |q_m| k_i^{(m)} \right) I \tag{4.66}$$

then $A \succeq 0$.

For a Linear Matrix Inequality, we cannot directly implement a decision variable value $|q_m|$. However, by introducing a slack variable α_m with

$$\alpha_m + q_m \geq 0, \alpha_m \geq 0 \tag{4.67}$$

then we know that $2\alpha_m + q_m \geq |q_m|$. The variable α_m will automatically be minimised in the SDP implementation if required. Thus in implementing semi-definite programming, we use the inequality condition

$$A_{ii} \succeq \left(k_i^{(0)} + \sum_m (2\alpha_m + q_m) k_i^{(m)} \right) I \tag{4.68}$$

along with (4.65) and (4.67).

4.4.5 Algorithms

The perturbation methods for invariant SDPs and SOS result in algorithms for the SDPs.

Algorithm A - Upper and Lower Bounds

Input: Parameterised matrices of subspace $\mathcal{L} \cap \mathbf{S}^N$, Linear representation σ of group G, matrices R and W for transformation Φ.

1. Find parameterised matrix of invariant subspace $\mathcal{U} \cap \mathbf{S}^N$.

2. Find parameterised matrix of invariant subspace $\mathcal{B} \cap \mathbf{S}^N$, where $\Phi(\bar{Q}) = R + W \circ \bar{Q}$.

Output Parameterised matrices $\bar{Q}$, Q_F of subspace $\mathcal{L} \cap \mathbf{S}^N$.

Necessary conditions and a lower bound can be found from $\bar{Q}$ while sufficient conditions and an upper bound can be found from Q_F.

Algorithm B - Block Diagonalisation

Input: Parameterised matrices $\bar{Q}$, Q_F, Orthogonal linear representation ρ of group G which induces representation σ, Transformation H, K.

- Find a Symmetry Adapted Transformation T for subspace $\mathcal{U} \cap \mathbf{S}^N$.
- Find parameterised block diagonal matrix $Q_D = T^T \bar{Q} T$.
- Find $Q_H = H(T, Q_F)$ for general case or $Q_H = T^T W^T Q_F W T$ for the simplest weighted case.

Output Parameterised Matrix Q_H, Q_D.

Necessary conditions can be found by testing Q_D and sufficient conditions from Q_H.

4.5 Networked Polynomials using S_N

We look at the case where the coupling term is of the same form and so analyse polynomials of the form

$$p(x_1, \ldots, x_N) = \sum_{i=1}^{N} p_i(x_i) + \sum_{i=1}^{N} \sum_{k=1}^{N} b_{ik} p_c(x_i, x_k) \tag{4.69}$$

where for simplicity we assume that $p_c(x_i, x_j) = p_c(x_j, x_i)$ and $b_{ij} = b_{ji}$.

For the case of a structured polynomial, we can apply the symmetry group S_N. If

$$p(x_1, \ldots, x_N) = \sum_{i=1}^{N} p_i(x_i) + \sum_{i=1}^{N} \sum_{k=1}^{N} b_{ik} p_c(x_i, x_k) \text{ is SOS} \tag{4.70}$$

then

$$p_{av}(x) = \frac{1}{N} \sum_{i=1}^{N} p_n(x_i) + \sum_{i=1}^{N} \sum_{k=1}^{N} b_{av} p_c(x_i, x_k) \text{ is SOS} \tag{4.71}$$

where $p_n = \sum_{i=1}^{N} p_i(x_i)$. This results from applying representation θ of S_N to $p(x)$ using Theorem 4.18.

In order to use this, we first determine the components in the gram matrices. We have

$$p_i(x_i) = Z_i(x_i)^T P_a Z_i(x_i) \tag{4.72}$$

for $i = 1, \ldots, N$ where $Z_i(x_i) \in \mathbb{R}^m$ is a vector of monomials and the symmetric matrix $P_i = P_i(\lambda_i) \in \mathbb{R}^{m \times m}$ is affinely parameterised by decision variables λ_i as in typical Gram matrix representations. Furthermore, we write

$$p_c(x_i, x_k) = Z_c(x_i, x_k)^T Q_c Z_c(x_i, x_k) \tag{4.73}$$

for $i, k = 1, \ldots, N$, where $Z_c(x_i, x_k)$ is a vector of monomials in x_i and x_k and $Q_c = Q_c(\lambda_c)$ is affinely parameterised by decision variables λ_c as in typical Gram matrix representations.

$$Q_c = \begin{bmatrix} Q_c^{11} & Q_c^{12} & Q_c^{13} \\ Q_c^{21} & Q_c^{22} & Q_c^{23} \\ Q_c^{31} & Q_c^{32} & Q_c^{33} \end{bmatrix}, Z_c(x_i, x_k) = \begin{bmatrix} Z_i \\ Z_k \\ Z_{ik} \end{bmatrix} \tag{4.74}$$

To analyse a 4th order polynomial where $x_i \in \mathbb{R}$ for $i = 1, \ldots, N$, the standard vectors of monomials are

$$Z_i(x_i) = (x_i, x_i^2), Z_{ik} = (x_i x_k) \tag{4.75}$$

From these monomial vectors Q_c is parameterised using the dependencies between monomials of $(x_i^2)(x_k^2) = (x_i x_k)^2$, $(x_i^2)(x_k) = (x_i)(x_i x_k)$ and $(x_k^2)(x_i) = (x_k)(x_i x_k)$. We also have the two dependencies $(x_i x_j)(x_j x_k) = x_j^2(x_i x_k)$ and $(x_i x_j)(x_k x_l) = (x_i x_k)(x_j x_l) = (x_i x_l)(x_j x_k)$, where the last cancels for the symmetric solution.

Necessary Conditions - S_N

If the polynomial (4.69) is SOS then there exist a matrix $\bar{Q}$ such that

$$\bar{Q} = \left[\begin{array}{c|c} \bar{Q}_1 & \bar{Q}_2 \\ \hline \bar{Q}_2^T & \bar{Q}_3 \end{array}\right] \succeq 0 \tag{4.76}$$

where

$$\begin{aligned} \bar{Q}_1 &= a_{av}(I_N \otimes P_n) + k_{av}(I_N \otimes 2\bar{Q}_c^{11}) + B_{av} \otimes 2\bar{Q}_c^{12} \\ \bar{Q}_2 &= b_{av}\bar{L}_B \otimes 2\bar{Q}_c^{13} + \bar{L}_B^c \otimes Q_j \\ \bar{Q}_3 &= b_{av} I_N \otimes \bar{Q}_c^{33} + \bar{\Omega} \otimes Q_f \end{aligned} \tag{4.77}$$

where $\bar{Q}_c^{11} = \bar{Q}_c^{22}$, $Q_f = q_f$, $Q_j = \begin{bmatrix} 0 \\ -q_f \end{bmatrix}$, $\bar{B}$ is the averaged adjacency matrix using S_N, $\bar{L}_B$ is the unweighted incidence matrix, L_B^c is the complement to the unweighted incidence matrix such that $\bar{L}_B + \bar{L}_B^c = 1_{N,L}$, $\Omega_{ij} = 1$ if edge i and edge j have a common node and $i \neq j$, and $\Omega_{ij} = 0$ otherwise.

Sufficient Conditions - S_N

The polynomial (4.69) is a Sum of Squares if there exist matrix Q_F such that

$$Q_F = \left[\begin{array}{c|c} Q_1 & Q_2 \\ \hline Q_2^T & Q_3 \end{array}\right] \succeq 0 \tag{4.78}$$

where,

$$\begin{aligned} Q_1 &= diag(a_1 P_n, \ldots, a_n P_n) + diag(2k_1\bar{Q}_c^{11}, \ldots, 2k_n\bar{Q}_c^{11}) + B \otimes 2\bar{Q}_c^{12} \\ Q_2 &= L_B \otimes 2\bar{Q}_c^{13} + L_B^c \otimes Q_j \\ Q_3 &= diag(b_{12}\bar{Q}_c^{33}, \ldots, b_{N,N-1}\bar{Q}_c^{33}) + \Omega \otimes Q_f \end{aligned} \tag{4.79}$$

where $B = [b_{ij}]$ is the adjacency matrix, L_B is the weighted incidence matrix, $\Omega_{ij} = \alpha_{ij}$ if edge i and edge j have a common node and $i \neq j$, and $\Omega_{ij} = 0$ otherwise. α_{ij} are free parameters which can be chosen to match the transformations K and H in 4.57. L_B^c is the complement to the unweighted incidence matrix such that $L_{B,ij} \neq 0$ or $L_{B,ij}^c \neq 0$. The

non-zero elements of L_B^c equal with the corresponding α_{ij}.

Block Diagonalisation - S_N

We can also block diagonalise $\bar{Q}$ and approximately block diagonalise Q_F using

$$T = \left[\begin{array}{cccccc} T_{n,1} & 0 & \dots & T_{n,N} & 0 & 0 \\ \hline 0 & \bar{L}^T T_{n,1} & \dots & 0 & \bar{L}^T T_{n,N} & V \end{array}\right] \tag{4.80}$$

where $\bar{L}$ is the unweighted incidence matrix, V is a set of orthonormal column vectors such that $LV = 0$ and

$$T_{n,1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} T_{n,2} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \end{bmatrix} T_{n,3} = \begin{bmatrix} 1 \\ 1 \\ -2 \\ 0 \\ \vdots \end{bmatrix} T_{n,N} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -(N-1) \end{bmatrix} \tag{4.81}$$

Computational Complexity - S_N

We next look at the computational complexity involved with introduced techniques. To analyse the computational complexity, we need to see how both the size of the matrices and the number of decision variables in the SDP scale with the size of the network. This is important as the flop cost of interior point methods is approximately linear with respect to the size of the LMI and polynomial with respect to the number of decision variables [7].

If we analyse (4.69) without using its structure to reduce the size of the SDP, then we have

$$\text{Size of Matrix} = O(L) \times O(L) = O(N^2) \times O(N^2) \tag{4.82}$$

$$\text{Number of Dec. Var.} \ = O(L^2) = O(N^4) \tag{4.83}$$

where L is the number of links in the network and N is the number of nodes. For this

case, it can be seen that as the size of the network increases, the system does note scale well. In particular, the number of decision variables has quartic scaling compared to the size of the network. As the computational time scales approximately polynomialy with respect to the number of decision variables, the computational time scales badly with respect to the size of the network.

By using the perturbation method without block diagonalistion, we obtain the scalability

$$\text{Size of Matrix} = O(L) \times O(L) = O(N^2) \times O(N^2) \tag{4.84}$$

$$\text{Number of Decision Variables } = O(1) \tag{4.85}$$

We can see that the size of the matrices increases and the number of decision variables remains constant as the number of links in the network increases. Thus the number of decision variables is no longer a scalability issue. But the computational time scales polynomially due to the size of the matrices, and so the size of the matrices is still a scalability constraint

If we use block diagonalisation we can further improve scalability by breaking one large matrix into many small matrices

$$\text{Number of Matrices} = O(L) = O(N^2) \tag{4.86}$$

$$\text{Size of each Matrix} = O(1) \times O(1) \tag{4.87}$$

$$\text{Number of Decision Variables } = O(1) \tag{4.88}$$

and so the system scales well with the size of the network.

Thus we have significantly improved the scalability through the number of decision variables and the size of the matrices by perturbing a symmetric solution. This scalability occurs from the reduction in the number of decision variables due to averaging, as well as the reduction in the size of matrices due to block diagonalisation.

4.6 Examples

In the following section we give details of structured Sum of Squares computations which were carried out to analyse 4th order polynomials with a network structure.

We analysed polynomials with a globally coupled network structure. The technique was tested using networks with node and link weightings which were not identical. The CPU time t_{cpu} and Wall Clock time t_{clock} for the computations were recorded and, as the network weightings were random, the times were averaged over a number of runs. The SDP solution was carried out using SDPT3 [73] or SeDuMi 1.3 [80] on MATLAB (R2013) through a YALMIP interface [52] on a 2.66GHz Intel Core i7-M620 CPU (dual core).

4.6.1 Non negativity of Polynomials

Example 4.29 *Consider*

$$p = \sum_{i=1}^{N} a_i(x_i^2 + x_i^4) - \sum_{i=1}^{N}\sum_{k=1}^{N} b_{ik} x_i^2 x_k^2 \tag{4.89}$$

where $x_i \in \mathbb{R}$ *for all* i. *Now*

$$Q_n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.90}$$

and

$$\bar{Q}_c = \left[\begin{array}{cc:cc:c} 0 & 0 & 0 & q_{c2} & -q_{c2} \\ 0 & 0 & q_{c2} & -q_{c1} & 0 \\ \hdashline 0 & q_{c2} & 0 & 0 & -q_{c2} \\ q_{c2} & -q_{c1} & 0 & 0 & 0 \\ \hdashline -q_{c2} & 0 & -q_{c2} & 0 & -1+2q_{c1} \end{array}\right]$$

where (q_{c1}, q_{c2}) *are variables in the SDP. Thus*

$$Q_c^{11} = Q_c^{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, Q_c^{12} = \begin{bmatrix} 0 & q_{c2} \\ q_{c2} & -q_{c1} \end{bmatrix} \tag{4.91}$$

$$Q_c^{13} = \begin{bmatrix} -q_{c2} \\ 0 \end{bmatrix}, Q_c^{23} = \begin{bmatrix} -q_{c2} \\ 0 \end{bmatrix}, Q_c^{33} = [-1 + 2q_{c1}] \tag{4.92}$$

We set W *to be a weighted diagonal matrix where* $W_{2i-1} = W_{2i} = a_i/a_{av}$ *for* $i = 1, \ldots, N$ *and* $W_i = \hat{b}_k/\hat{b}_{av}$ *for* $i = 2N + 1, \ldots, 2N + L$.

The technique was tested on a globally coupled network using randomly chosen weighting of $0.5 \leq a_i \leq 1.5$ *and* $0 \leq b_{ij} \leq \frac{1}{N}$.

During scalability testing, it was found that an SOS for $N = 60$ *could be determined by the SDP in 2 min for a globally coupled network using Algorithm A. Using Algorithm A and B, it was found that an SOS for* $N = 100$ *could be determined with preprocessing time of 2 min and the SDP time of approx.* $3s$. *The preprocessing step for algorithm A and B together contains two bottlenecks, the multiplication* $T^T Q_F T$ *and parsing in YALMIP. In future work, the first bottleneck should be possible to remove using parallel processing for matrix multiplication, as the computations were only completed on 2 cores, rather than multiple cores or a GPU. The YALMIP bottleneck should be possible to remove by using the SDP solvers directly.*

For both of the above problems existing Sum of Squares toolboxes were able to verify SOS for $N = 15$, *but were computationally impractical for* $N = 20$, *by which we mean that the computational time taken was significantly greater than 10 minutes and scaled badly (high order polynomial) with increasing network size.*

For repeated trials of Algorithm A and B together when $N = 30$, *the sufficiency test held the majority of the time when* $diag(a_1, \ldots, a_N) - B \succeq 0$. *The technique was more conservative for Algorithm A and B together for larger values of* N.

Example 4.30 *Consider*

$$p = \sum_{i=1}^{N} a_i(x_i^2 + x_i^4) + \sum_{i=1}^{N}\sum_{k=1}^{N} b_{ik}(x_i^4 - 3x_i^3x_j + 3x_i^2x_j^2 - 3x_i^3x_j + x_i^4 - x_ix_j^2 - x_i^2x_j) \quad (4.93)$$

where $x_i \in \mathbb{R}$ *for all* i*. Now*

$$Q_n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.94)$$

and

$$\bar{Q}_c = \left[\begin{array}{cccc:c} 0 & 0 & 0 & -0.5+q_{c2} & -q_{c2} \\ 0 & 1 & -0.5+q_{c2} & 1.5-q_{c1} & -1.5 \\ \hdashline 0 & -0.5+q_{c2} & 0 & 0 & -q_{c2} \\ -0.5+q_{c2} & 1.5-q_{c1} & 0 & 1 & 0 \\ \hdashline -q_{c2} & -1.5 & -q_{c2} & -1.5 & 2q_{c1} \end{array}\right]$$

where (q_{c1}, q_{c2}, q_{c2}) *are variables in the SDP. We set* W *to be a weighted diagonal matrix where* $W_{2i-1} = W_{2i} = a_i/a_{av}$ *for* $i = 1, \ldots, N$ *and* $W_i = \hat{b}_k/\hat{b}_k$ *for* $i = 2N+1, \ldots, 2N+L$*. The weighting equates to scaling the 'node' monomials by the 'node' parameter* a_i *while scaling the 'edge' monomials by the 'edge' parameter* $\hat{b}_k$*.*

The technique was tested on a globally coupled network using randomly chosen weighting of $2 \leq a_i \leq 4$ *and* $\frac{1}{N} \leq b_{ij} \leq \frac{2}{N}$*. For this case, Algorithm A showed that* $p(x)$ *is SOS in the majority of test cases. Algorithm A and B together with* W *chosen as in Example 4.29 was typically found to be conservative when compared to Algorithm A.*

Scalability for both Algorithm A and Algorithm A and B together was found to be near identical to the previous example.

4.6.2 Lyapunov functions with a Network Structure

We can use the technique to analyse networked systems with naturally structured Lyapunov function candidates, such as 'Damped Hamiltonian' or Gradient systems. Many

types of networks fit into these classes of system, such as power systems [16], synchronization models [33, 40], population dynamics [72] and neural networks [41, 44]. The advantage of systems of this form is that there exists a natural Lyapunov function candidate which is often highly structured and whose derivative with respect to time is typically globally non-positive. Thus if we can show that the Gradient or Hamiltonian function is positive definite then we can show that the system is stable.

An example of this case is a network of modified Duffing oscillators [41], where the coupling has been added in the form of a the network. This can be written with 3$^{\text{rd}}$ order polynomial coupling as

$$\begin{aligned} \dot{z}_i &= -\gamma_i z_i - a_i(y_i - y_i^3) + \sum_{j=1}^{N} b_{ij}(y_j - y_i)^3 \\ \dot{y}_i &= z_i \end{aligned} \tag{4.95}$$

where $i = 1, \ldots, N$, $b_{ij} = b_{ji}$ and $b_{ii} = 0$. If we let

$$V = \sum_{i=1}^{N} \left[\frac{1}{2} z_i^2 + a_i \left(\frac{1}{2} y_i^2 - \frac{1}{4} y_i^4 \right) \right] + \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} b_{ik} \frac{1}{4} (y_k - y_i)^4 \tag{4.96}$$

$$\begin{aligned} \dot{V} =& \sum_{i=1}^{N} (z_i(-\gamma_i z_i - a_i(y_i - y_i^3)) + a_i(y_i - y_i^3) z_i) \\ &+ \sum_{i=1}^{N} z_i \sum_{j=1}^{N} b_{ij}(y_j - y_i)^3 - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} 2 b_{ik}(y_k - y_i)^3 z_i \\ =& - \sum_{i=1}^{N} \gamma_i z_i^2 \leq 0 \end{aligned} \tag{4.97}$$

using $b_{ij} = b_{ji}$.

$$\dot{V} = - \sum_{i=1}^{N} \gamma_i z_i^2 \leq 0 \tag{4.98}$$

We can see by observation that $-\dot{V}$ is a Sum of Squares and so we only need to show that V is positive definite for a domain containing the origin to show that the origin is stable [41]. We next analyse (4.96) using perturbation methods for invariant polynomials.

Example 4.31 *Testing whether* (4.96) *is positive definite can be reduced to testing whether*

$$V_1 = \sum_{i=1}^{N} a_i(\frac{1}{2}y_i^2 - \frac{1}{4}y_i^4) + \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} b_{ik}\frac{1}{4}(y_k - y_i)^4 \text{ is SOS} \tag{4.99}$$

This is locally positive definite about $y_i = 0$ *but we require techniques to find whether it is positive definite for a larger domain. This an important starting point for analysing non-linear behaviour such as finding the region of attraction estimate if the equilibrium point is asymptotically stable.*

To find the positive definite domain, we tested whether

$$p(y_1, \ldots, y_N) = V_1 - \sum_{i=1}^{N} \phi_i(y_i) \text{ is SOS} \tag{4.100}$$

where

$$\phi_i(y_i) = \lambda_i y_i^2(g - y_i^2) \tag{4.101}$$

and $\lambda_i > 0$*, which, when true, implies that* V_1 *is positive definite for* $y_i^2 < g$*.*

In this case, in the notation of Section 4.5,

$$p_i = a_i(\frac{1}{2}y_i^2 - \frac{1}{4}y_i^4) - \lambda b_i y_i^2(g - y_i^2), i = 1, \ldots, N \tag{4.102}$$

$$p_c = \frac{1}{2}(0.25y_k^4 - y_k^3 y_i + 1.5y_i^2 y_k^2 - y_k y_i^3 + 0.25y_i^4) \tag{4.103}$$

$$P_i = a_i \begin{bmatrix} 0.5 & 0 \\ 0 & -0.25 \end{bmatrix} - \lambda_i \begin{bmatrix} g & 0 \\ 0 & -1 \end{bmatrix}$$

where $\lambda_i > 0$ *are parameters which are not due to dependencies in the monomial vector*

and which are used as decision variables in the SDP. Also,

$$Q_c = \frac{1}{2}\left[\begin{array}{cc:cc:c} 0 & 0 & 0 & q_{c2} & -q_{c2} \\ 0 & 0.25 & q_{c2} & -q_{c1} & -0.5 \\ \hdashline 0 & q_{c2} & 0 & 0 & -q_{c2} \\ q_{c2} & -q_{c1} & 0 & 0.25 & -0.5 \\ \hdashline -q_{c2} & -0.5 & -q_{c2} & -0.5 & 1.5+2q_{c1} \end{array}\right]$$

where $\lambda_c = (q_{c1}, q_{c2}, q_{c2})$ *are decision variables in the SDP. Thus*

$$Q_c^{11} = Q_c^{22} = \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{4.104}$$

$$Q_c^{12} = \frac{1}{2}\begin{bmatrix} 0 & q_{c2} \\ q_{c2} & -q_{c1} \end{bmatrix}, Q_c^{13} = \frac{1}{2}\begin{bmatrix} -q_{c2} \\ -0.5 \end{bmatrix} \tag{4.105}$$

$$Q_c^{23} = \frac{1}{2}\begin{bmatrix} -q_{c2} \\ -0.5 \end{bmatrix}, Q_c^{33} = \frac{1}{2}[1.5 + 2q_{c1}] \tag{4.106}$$

For the globally coupled network we set $g = 1.8$ *and we randomly set* $\frac{0.5}{N} \leq b_{ij} \leq \frac{1.5}{N}$ *and* $0.5 \leq a_i \leq 1.5$.

For a globally coupled network it was possible to analyse a system using Algorithm A with $N = 60$ *in 2 min.*

For the same problem existing Sum of Squares toolboxes were able to verify SOS for $N = 15$*, with* $N = 20$ *computationally impractical. For test cases near* $a_i = a_{av}, b_{ij} = b_{av}$*, there was little conservatism.*

4.7 Future Directions

There are a number of future directions to develop the methodology, including improved testing of accuracy and bounds, using different group representations, using sets of transformations which do not have a group structure, and using convex combinations of generated matrices for the perturbation. Finally, once the methodology for particular transformations has been fully developed, techniques to find the appropriate transformations given an arbitrary SDP or polynomial need to be found.

The next step in testing of the method is to use the necessary and sufficient conditions to find the accuracy of the method, rather than testing sufficient conditions on set cases. This will be both for feasibility as well as the SDP bounds.

Different group representations need to be applied, which will reduce conservatism for certain classes of SDP or polynomials. As an example, the dihedral group can be used for polynomials which are perturbations of a lattice structure.

Another future direction of research is to use sets of transformations without a group structure to give upper and lower bounds of an SDP, as well as necessary and sufficient conditions for feasibility. The group structure is required for block diagonalisation, but is not required for the averaging method, which only relies on convexity. The group structure is not actually required for Lemma 4.16 or subsequent theorems in this section, as the average can be obtained for any finite set of transformations $\{f_i(X), i = 1, \ldots, p\}$ which preserve symmetric positive semi-definite matrices. For

$$\mathcal{U} = \{\bar{X} \in \mathbf{S}^N | X \in \mathcal{L}\}, \bar{X} = \frac{1}{p} \sum_{i=1}^{p} f_i(X) \tag{4.107}$$

then $X \in \mathcal{L} \cap \mathbf{S}^N_+$ implies $f_i(X) \in \mathcal{U} \cap \mathbf{S}^N_+$ for $i = 1, \ldots, p$, using convexity. Thus we can generate a new space using $\{f_i\}$. However, without the group structure, $\bar{X}$ is not necessarily an invariant matrix. This invariance of $\bar{X}$ is required for block diagonalisation, and so using this general set of transformations only allows Algorithm A, but will potentially

give a smaller 'gap'.

Using convex combinations of generated matrices and polynomials is an interesting future direction of research. One method to reduce conservatism between necessary and sufficient conditions or upper and lower bounds is to generate multiple invariant SDPs, which can then be computed in parallel. For the case of SOS, if $p(x)$ is SOS, then for representations $\theta^{(1)}(g)$ and $\theta^{(2)}(g)$ there must exist SOS polynomials $\bar{p}^{(1)}(x)$ and $\bar{p}^{(2)}(x)$. Necessary conditions for all representations must hold, and testing multiple conditions may reduce the conservatism of the necessary conditions. Also, although this increases computational complexity, the multiple generated necessary conditions can be computed in parallel and so the elapsed real time of computations will not necessarily increase.

For sufficient conditions and upper bounds using convex combinations, if we perturb a convex combination of invariant subspaces (or polynomials) we can reduce the perturbation required and so reduce the gap between the invariant and actual problem. $\alpha\bar{p}^{(1)}(x) + (1-\alpha)\bar{p}^{(2)}(x)$ may be a smaller perturbation from $p(x)$ than $\bar{p}^{(1)}(x)$ or $\bar{p}^{(2)}(x)$ individually. If we perturb $\bar{p}^{(1)}(x)$ and $\bar{p}^{(2)}(x)$ appropriately, we can then test whether $p(x) = \alpha\bar{p}^{(1)}_{H_1}(x) + (1-\alpha)\bar{p}^{(2)}_{H_2}(x)$ is SOS by testing whether $\bar{p}^{(1)}_{H_1}(x)$ and $\bar{p}^{(2)}_{H_1}(x)$ are SOS. Once again, these may be computed in parallel and so the elapsed real time of computations will not necessarily increase. Interestingly, it may be better to use a combination of representations and standard transformation of SDPs or polynomials to achieve this. For example, $\frac{1}{2}\bar{p}^{(1)}(x) + \frac{1}{2}\bar{p}^{(1)}(Ax)$, where $\bar{p}^{(1)}$ is generated from a representation and A is a transformation, may further decrease the perturbation to $p(x)$.

Finally, once the methodology for particular representations has been fully developed, further techniques will be required to find the appropriate representations given an arbitrary SDP or polynomial. An example of this would be to use product groups for clustered systems, where each cluster has a different group representation. The development of finding the appropriate transformations given an arbitrary SDP will largely be dependent upon the direction of development of the above suggested research.

4.8 Conclusion

In this chapter we have introduced perturbation methods for invariant Sum of Squares and Semi-definite Programming. Using convexity and group representation theory, we have found a methodology which gives upper and lower bounds and necessary and sufficient feasability conditions for SDPs and SOS, with significant computational benefits over analysing the SDP or SOS directly. We have applied this technique to analyse two examples of large polynomial functions with a network structure, as well as a networked polynomial Lyapunov function. The technique was shown to be applicable for large problems of over 100 variables in SOS programming where typical SOS programming techniques are only feasible up to 15 variables.

Chapter 5

Invariance Principles for Time-varying Systems

5.1 Introduction

The Krasovskii-Lasalle Invariance Principle [15, 45–48], or Lasalle's Invariance Principle, is a fundamental tool for stability analysis of differential equations. The principle can be used to study the asymptotic behaviour of a system when a Lyapunov function's derivative is not negative definite [29, 41], which often occurs for the standard choice of Lyapunov function, e.g. the energy for a mechanical system [41]. It also has the other advantages of not requiring a positive definite Lyapunov function, providing a more general region of attraction estimate and giving the ability to study the attractivity of sets rather than just equilibrium points [41].

However, the invariance principle only holds for autonomous systems and special cases of time-varying systems [29, 41], and there is a lack of both simple and general results for the time-varying case. The invariance principle can be applied for the special cases of periodic, almost periodic or asymptotically independent systems [29], but not for more general cases. The standard proof of the invariance principle uses the invariance of the

limits sets, but for the time-varying case the limits sets are not necessarily invariant. Invariance-like principles for time-varying systems can be obtained by applying Barbalat's Lemma [14, 29, 78] or the Lasalle-Yoshizawa Theorem [89], [48]. However, these methods often cannot show that the origin is (uniformly) asymptotically stable. For example, it is possible to show that the velocity of a pendulum approaches zero using the Lasalle-Yoshizawa Theorem, but not that the position approaches zero.

A range of techniques have been developed in order to study invariance for the time-varying case beyond special cases, however, these are typically difficult to implement for general cases. Invariance principles can be applied under more general conditions using 'limiting equations', which generalise the concept of limit sets for non-autonomous systems [9–11]. Using a conceptually similar approach, a generalisation of the Krasovskii-Lasalle invariance principle for differential inclusions can also be used to treat a class of non-autonomous dynamical systems [54, 55]. There are other techniques for non-autonomous systems which require conditions, often related to observability and detectability, that ask for a Lyapunov function to always decrease over a period of time rather than at an instant in time [1–3] (also see references in [12]) and techniques which combine the above methods with limiting equations [12, 49, 50]. It is also possible to find stability conditions for time-varying systems using one or more auxiliary functions by applying Matrosov's Theorem [59], [74] (see [57] and references therein for more recent versions). If Lyapunov function derivatives are semi-definite and asymptotic stability cannot easily be shown, it is also possible to use non-Lyapunov function methodology, such as contraction analysis [56].

In this chapter we introduce relatively simple and general invariance principles for time-varying systems. We first provide a new proof of the Krasovskii-Lasalle Invariance Principle for forward time, by using the invariance properties of regions of attraction rather than the limit set. This gives a new perspective to the problem and shows that the concept of a limit set is not required for proofs of invariance principles. We then propose and prove invariance principles for time-varying systems, where we analyse trajectories

leaving sets. This can be viewed as a time-varying equivalent to Barbashin-Krasovskii Theorem [15, 45], where the analysis requires the introduction of trajectories leaving sets 'uniformly in t'. We also propose methods to analyse trajectories leaving sets using transformed coordinates, which simplifies analysis.

The proposed invariance principles simplify and extend a number of existing results for time-varying ordinary differential equations. The transformation of coordinates can be related to the auxiliary function in Matrosov's Theorem, or the output function in observability-type methods. When the transformation is viewed as an 'auxiliary' function, the method is simpler and more general than Matrosov's Theorem, due to simpler and less restrictive conditions on the auxiliary functions. When the transformation is viewed as an 'output' function, the proposed method is simpler to apply than observability conditions for time-varying systems, as conditions do not need to be verified along a system trajectory. The methodology is simpler than using limiting equations or differential inclusion as the principle applies to the vector field directly. For application to non-autonomous ordinary differential equations, the proposed principles are also more general than differential inclusions techniques as they allow for time-varying Lyapunov functions.

In this Chapter, Section 5.2 gives background on dynamical systems, Section 5.3 provides a new proof for Krasovskii-Lasalle's invariance principle, Section 5.4, introduces a framework for invariance in non-autonomous systems, Section 5.5 introduces and applies invariance principles for non-autonomous systems, and Section 5.6 gives applications to large-scale systems.

5.2 Background: Dynamical Systems and Invariance

We first introduce necessary background on dynamical systems and set stability.

5.2.1 Dynamical Systems

Recall that Equation (2.3) refers to the system $\dot{x} = f(x)$ with $x(0) = x_0$. Let $\phi(t, x_0)$ be the positive semi-flow of (2.3) where $\phi(t, x_0) = x(t)$ for $t \in \mathbb{R}^+$ and $x(0) = x_0$.

Recall from Chapter 2, that a set M is said to be positively invariant if $x_0 \in M$ implies $x(t) \in M$ for all $t \in \mathbb{R}^+$ and invariant if $x_0 \in M$ implies $x(t) \in M$ for all $t \in \mathbb{R}$.

Lemma 5.1 *[18] If a set M is (positively) invariant then $\overline{M}$ is also (positively) invariant. If sets M and P are (positively) invariant then $M \cap P$ and $M \cup P$ are (positively) invariant.*

Definition 5.2 *The system state $x(t) \to \Lambda$ as $t \to \infty$ if for every $\varepsilon > 0$ there exists $T \geq 0$ such that $d(x(t), \Lambda) \leq \varepsilon$ for all $t \geq T$.*

Notation: for the rest of the thesis, '$x \to \Lambda$' is used as shorthand for '$x(t) \to \Lambda$ as $t \to \infty$' and '$V(x) \to b$' as shorthand for '$V(x(t)) \to b$ as $t \to \infty$' unless otherwise stated.

Definition 5.3 *The positive limit set is the set of all points for which a sequence $\{t_n\}$ exists with $t_n \to \infty$ as $n \to \infty$ such that $x(t_n) \to p$ as $n \to \infty$.*

Lemma 5.4 *If a solution of (2.3) is bounded and belongs to D for all $t \geq 0$, then its positive limit set L^+ is a non-empty, compact, invariant set and $x(t) \to L^+$.*

Proof. See [41]. ■

5.2.2 Continuous Dependence on Initial Conditions: Time-varying systems

We next present a theorem which justifies continuous dependence of the solution of (2.1) on initial conditions for time-varying systems.

Recall that Equation (2.1) refers to system $\dot{x} = f(x, t)$ with $x(t_0) = x_0$.

Theorem 5.5 *Let $f(x, t)$ be piecewise continuous in t and Lipschitz continuous in x on $[t_0, t_0] \times W$ with Lipschitz constant L, where W is an open connected set. Let $y(t)$ and*

$z(t)$ be solutions of (2.1) with $y(t) = y_0$ and $z(t) = z_0$, such that $y(t), z(t) \in W$ for all $t \in [t_0, t_0]$. Then

$$\|y(t) - z(t)\| \leq \|y_0 - z_0\| e^{L(t-t_0)} \tag{5.1}$$

for all $t \in [t_0, t_0]$.

Proof. See [41]. ■

5.2.3 The Invariance Principle

We next state the invariance principle for autonomous systems.

Theorem 5.6 *[41, Theorem 3.4] Let $\Omega \subset D$ be a compact positively invariant set with respect to (2.3). Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V} \leq 0$ in Ω. Let E be the set of all points in Ω where $\dot{V} = 0$. Let M be the largest positively invariant subset in E. Then every solution starting in Ω approaches M as $t \to \infty$.*

Example 5.7 *Let*

$$\begin{aligned} \dot{z}_1 &= -\gamma z_1 - b \sin z_2 \\ \dot{z}_2 &= z_1 \end{aligned} \tag{5.2}$$

Let $z_0 \in \Theta_z = \{z \in \mathbb{R}^2 | \frac{1}{2} z_1^2 + b(1 - \cos z_2) \leq c, |z_2| \leq \pi\}$, where $c < 2$. Now using Theorem 5.6

$$V = \frac{1}{2} z_1^2 + b(1 - \cos z_2) \tag{5.3}$$

$$\dot{V} = -\frac{1}{2} \gamma z_1^2 \leq 0 \tag{5.4}$$

As $\dot{z}_1 \neq 0$ for $z_1 = 0$, $z_2 \neq 0$, then $\{(0, 0)\}$ is the largest invariant subset of E and so $(z_1, z_2) \to 0$.

5.2.4 Invariance-like Theorems for Time-varying Systems

There is not a direct equivalent to the invariance principle for time-varying systems outside special cases, as limit sets for time-varying systems are not necessarily invariant. Instead

weaker forms of attractivity can be shown when derivatives of Lyapunov functions are positive semi-definite.

In this section invariance-like theorems are provided for time-varying systems.

Theorem 5.8 *For System* (2.1)*, let* $\Theta_z \subset D_z$ *be a compact set which contains the origin and let* $V : D_z \times \mathbb{R}^+ \to \mathbb{R}^+$ *be a continuously differentiable function. Let* $w_1(z), w_2(z)$ *be continuous, positive definite functions and let* $w(z)$ *be a continuous, positive semidefinite function. If* $0 < c < \min_{z \in \partial\Theta_z} w_1(z)$*,* $z_0 \in \{z \in \Theta_z | w_2(z) \leq c\}$ *and*

$$w_1(z) \leq V(z,t) \leq w_2(z) \tag{5.5}$$

$$\dot{V}(z,t) \leq -w(z) \tag{5.6}$$

for all $(z,t) \in \Theta_z \times \mathbb{R}^+$ *then* $z \to E_z = \{z \in \Theta_z | w(z) = 0\}$*.*

Proof. $w(z(t)) \to 0$ follows from [41, Theorem 8.4], using a compact set rather than compact ball. ■

Example 5.9 *Let*

$$\begin{aligned} \dot{z}_1 &= -\gamma(t) z_1 - \beta(t) \sin z_2 \\ \dot{z}_2 &= z_1 \end{aligned} \tag{5.7}$$

where $0 < \beta_l \leq \beta(t) \leq \beta_h$ *is a continuous function,* $\gamma(t)$ *is a bounded continuous function and* $2\gamma(t)\beta(t) + \dot{\beta}(t) \geq p > 0$*. Let* $\Omega_z = \{z \in \mathbb{R}^2 | \frac{1}{2\beta_h} z_1^2 + (1 - \cos z_2) \leq 2, |z_2| \leq \pi\}$*,* $z_0 \in \{z \in \mathbb{R}^2 | \frac{1}{2\beta_l} z_1^2 + (1 - \cos z_2) \leq c, |z_2| \leq \pi\}$*, where* $c < 2$*. Let* $z_0 \in \Theta_z = \{z \in \mathbb{R}^2 | \frac{1}{2} z_1^2 + b(1 - \cos z_2) \leq c, |z_2| \leq \pi\}$*, where* $c < 2$*. Now using Theorem 5.8*

$$V = \frac{1}{2\beta(t)} z_1^2 + (1 - \cos z_2) \tag{5.8}$$

$$\dot{V} = -\frac{1}{2\beta(t)^2}(2\gamma(t)\beta(t) + \dot{\beta}(t)) z_1^2 \leq -\frac{p}{2\beta_h^2} z_1^2 \leq 0 \tag{5.9}$$

and so $z_1 \to 0$*. However, without an invariance principle, we cannot use Lasalle-Yoshizawa Theorem to show that* $(z_1, z_2) \to 0$*.*

In this chapter we are motivated to develop methods that can show that $(z_1, z_2) \to 0$ uniformly in t.

5.3 The Invariance Principle for Autonomous Systems

In this section we present a new proof of the Krasovskii-Lasalle Invariance Principle for forward time using invariance properties of the region of attraction rather than the invariance of the limit set. This alternative proof is of the same level of complexity, if not simpler than, the traditional proof of the Krasovskii-Lasalle Invariance principle together with proofs of the limit set properties.

Definition 5.10 *Region of Attraction (ROA)*

Let $V : D \to \mathbb{R}^+$ be a continuous function, let $\Theta \subset D$ be a closed, positively invariant set and let $R_{V,b}$ be defined as the set

$$R_{V,b} = \{x_0 \in \Theta | V(\phi(t, x_0)) \to b \text{ as } t \to \infty\}. \tag{5.10}$$

We call this set the Region of Attraction on Θ for which $V(x) \to b$.

We next prove an invariance property for functions approaching a limit, which is closely related to the invariance of a region of attraction [41] and positive invariance of forward trajectories [18].

Lemma 5.11 *Positive Invariance of the Region of Attraction*

For system (2.3) with non-empty $R_{V,b}$, then $R_{V,b}$ is positively invariant.

Proof. [41], [18] If $x_0 \in R_{V,b}$ then $V(x) \to b$. Now $x(t) = \phi(t, x_0)$ for $t \in \mathbb{R}^+$ and $V(\phi(s+t, x_0)) = V(\phi(s, \phi(t, x_0))) = V(\phi(s, x(t)))$ for all $s, t \geq 0$ and so $V(\phi(s, x(t))) \to b$ as $s \to \infty$ for all $x(t) \in R_{V,b}$. Thus $x(t) \in R_{V,b}$ for all $t \geq 0$. ■

We next prove the Krasovskii-Lasalle Invariance Principle.

Theorem 5.12 *The Krasovskii-Lasalle Invariance Principle for Forward Time*

Let $\Theta \subset D$ be a compact, positively invariant set, $x_0 \in \Theta$ and $V : D \to \mathbb{R}^+$ be a continuously differentiable function. Let $E = \{x \in \Theta | \dot{V}(x) = 0\}$ and let M be the largest positively invariant subset of E. If

$$\dot{V}(x) \leq 0 \text{ on } \Theta \tag{5.11}$$

then $x \to M$.

Proof.

As Θ is compact and V is continuous then there exists some constant a such that $V(x) \geq a$ on Θ. V is bounded from below and non-increasing for $x \in \Theta$ and so V tends to a limit $b = b(x_0) \geq a$ as $t \to \infty$ for all $x_0 \in \Theta$, noting that the value of b can depend upon x_0. Thus by definition, there exists a $b = b(x_0)$ and corresponding non-empty $R_{V,b}$ for all $x_0 \in \Theta$.

We first show that $x \to \overline{R}_{V,b} \bigcap L_{V,b}$ where $L_{V,b} = \{x \in \Theta | V(x) = b\}$. $V(x)$ has a minimum $V_{min} \geq b$ on $\overline{R}_{V,b}$ as V is continuous and $\overline{R}_{V,b}$ is compact. As $V(x) \to b$ and $\dot{V} \leq 0$ then the minimum is b and so $\overline{R}_{V,b} \bigcap L_{V,b}$ is non-empty. If we assume that $x \not\to \overline{R}_{V,b} \bigcap L_{V,b}$ then there exists $\delta > 0$ and $\{t_i\}$ where $t_{i+1} > t_i$, $t_i \to \infty$ as $i \to \infty$ such that $d(x(t_i), \overline{R}_{V,b} \bigcap L_{V,b}) > \delta$ for all i. Thus $x(t_i) \in \Lambda = \overline{R}_{V,b} \backslash N_\delta(\overline{R}_{V,b} \bigcap L_{V,b})$ for i. Now $\overline{\Lambda} \subset \Theta$ is compact and so V has a minimum on $\overline{\Lambda}$, where $V(\overline{x}) = V_{min} \geq b$. Now $V(x(t_i)) \to b$ as $i \to \infty$ and $\dot{V} \leq 0$ and so $V_{min} = b$. But we can choose $v \in \Lambda$ arbitrarily close to any $\overline{x} \in \overline{\Lambda}$ implying $d(v, \overline{R}_{V,b} \bigcap L^0_{V,b}) \leq \delta$, which contradicts the assumptions. Therefore $x \to \overline{R}_{V,b} \bigcap L_{V,b}$.

Now $\overline{R}_{V,b}$ is positively invariant (See Chapter 2) and so $\overline{R}_{V,b} \bigcap L_{V,b}$ is positively invariant as $\dot{V} \leq 0$. Thus if $x_0 \in \overline{R}_{V,b} \bigcap L_{V,b}$ then $V(x(t)) = b$ for all $t \geq t_0$ and so $\dot{V}(x) = 0$ on $\overline{R}_{V,b} \bigcap L_{V,b}$. Therefore $\overline{R}_{V,b} \bigcap L_{V,b} \subset M$ and so $x \to M$.

■

Example 5.13 *[41] For Example 5.9, with constant b, γ, for $x_1 = 0$ then $\dot{x}_1 = -\beta \sin x_2 \neq 0$ for $x_2 \neq 0$. Thus by Theorem 5.12, $M = \{(0,0)\}$ is the largest positively invariant set in E and $(x_1, x_2) \to (0,0)$.*

Theorem 5.12 differs from the classical result in that M is the largest positively invariant subset of E rather than the largest invariant subset, which is why it is referred to as the Krasovskii-Lasalle Invariance Principle 'for forward time'. This shows a minor contrast between using limit sets and regions of attraction for the proof.

In virtually all applications, such as the standard example of using the mechanical energy to analyse the stability of a damped pendulum in Example 5.13, there is no difference as the two versions are equivalent for forward time.

The Barbashin-Krasovskii Theorem is equivalent to positive invariant M [15, 41]. However, in Theorem 5.12, defining M as the largest positively invariant subset does not give us information when the dynamics of the system on E are uncoupled from $\dot{V}$, as can be seen in the following example.

Example 5.14

$$\begin{aligned} \dot{x}_1 &= -2x_1 - x_2 \\ \dot{x}_2 &= -x_1 - 2x_2 \end{aligned} \tag{5.12}$$

Where $\Theta = \{-c \leq x_1 + x_2 \leq c, -c \leq x_1 - x_2 \leq c\}$. Using $V(x) = \frac{1}{2}(x_1 - x_2)^2$ then $\dot{V} = -(x_1 - x_2)^2$. The largest positively invariant subset of E is $\{x \in \Theta | x_1 = x_2\}$ while the largest invariant subset is $\{x_1 = x_2 = 0\}$, as x leaves Θ for backwards time when $x_0 \neq 0$. This can be seen more clearly by setting $z_1 = x_1 - x_2$ and $z_2 = x_1 + x_2$, where $\Theta_z = \{-c \leq z_1 \leq c, -c \leq z_2 \leq c\}$

$$\dot{z}_1 = -z_1, \dot{z}_2 = -3z_2 \tag{5.13}$$

and where $V = \frac{1}{2}z_1^2$ *and* $\dot{V} = -z_1^2$. *Thus the largest positively invariant set is* $\{z \in \Theta_z | z_1 = 0\}$ *while the largest invariant set is* $\{z_1 = z_2 = 0\}$. *Interestingly, if we use equivalent observability conditions [41] and define an output function* $y = x_1 - x_2$ *so that* $\dot{V} = -\|y\|^2$ *then we note that the system is not observable and so we similarly cannot use observability to determine attractivity with* $V(x) = \frac{1}{2}(x_1 - x_2)^2$.

5.4 Trajectories leaving Sets Uniformly

In this section, we provide a number of definitions and properties of trajectories leaving sets uniformly, which give the foundation for analysing asymptotic behaviour of systems with non-increasing Lyapunov functions. This foundation is used in later sections to develop methodology and applications.

We first analyse trajectories leaving sets uniformly in the original states followed by in transformed coordinates. We also provide foundation Lyapunov theory necessary for analysis using non-increasing Lyapunov functions. It can be noted that the transformation of coordinates of trajectories is used in Section 5.5 to relate the methods to output (observability) and auxiliary function (Matrosov) methods for non-autonomous systems.

Recall that for System (2.1) then $\dot{x} = f(x,t)$. We assume throughout the rest of the chapter that $f(x,t)$ is bounded on the set of interest. Now if $f(0,t) = 0$, $\Omega_x \subset \mathbb{R}^n$ is compact, and $x(t) \in \Omega_x$ for all $t \geq 0$ then $f(x,t)$ is bounded as $f(x,t)$ is (locally) Lipschitz, uniformly in t.

5.4.1 Definitions and Properties of Trajectories leaving a Set

We first provide definitions and properties of trajectories leaving sets uniformly.

We define trajectories leaving sets uniformly in terms of time taken for a trajectory to reach a specified distance away from the set

Definition 5.15 *For $S \subset P \subset D$ and $\varepsilon : P \to \mathbb{R}$ where $\varepsilon > 0$, we say that (x,t) leaves P by $\varepsilon(x(t_0),t_0)$, uniformly in t, for all $(x(t_0),t_0) \in S$ if for all $(x(t_0),t_0) \in S$ there exists $T > 0$ such that for all $t_0 \geq 0$ then $d\big((x(t_0+t'),t_0+t'),P\big) > \varepsilon$ for some $0 < t' \leq T$.*

In Definition 5.15, we use the term 'uniformly in t' as T is independent of (x_0,t_0) and note that the definition gives uniformity in terms of t but not in terms of the distance from the set.

We next give definitions for trajectories leaving sets, which are uniform with respect to both t and the distance from the set.

Definition 5.16 *We say that (x,t) leaves P uniformly for all $(x_0,t_0) \in S$ if there exists $\varepsilon_m, T_m > 0$ such that for all $(x_0,t_0) \in S$ then $d_x\big((x(t_0+t'),t_0+t'),P\big) > \varepsilon_m$ for some $0 \leq t' \leq T_m$. We say that (x,t) leaves P uniformly for all (x_0,t_0) in a uniform neighbourhood of S if there exists $\varepsilon_S, \delta_S, T_S > 0$ such that for all $(x_0,t_0) \in N^x_{\delta_S}(S)$ then $d_x\big((x(t_0+t'),t_0+t'),P\big) > \varepsilon_S$ for some $0 \leq t' \leq T_S$.*

In the next two lemmas, we present some properties of trajectories leaving sets uniformly. The first Lemma presented shows the relationship between trajectories leaving a set P uniformly for $(x_0,t_0) \in S$ compared with (x_0,t_0) in a uniform neighbourhood of S.

Lemma 5.17 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$, and let $S \subset P \subset \Omega_x \times \mathbb{R}^+ \subset D$. (x,t) leaves P uniformly for all $(x_0,t_0) \in S$ if and only if (x,t) leaves P uniformly for all (x_0,t_0) in a uniform neighbourhood of S.*

Furthermore, if (x,t) leaves P uniformly for all $(x_0,t_0) \in S$ then for the uniform neighbourhood of S we can set $0 < b < 1$ such that $\delta_S = b\varepsilon_m e^{-LT_m}$, $\varepsilon_S = (1-b)\varepsilon_m$, $T_S = T_m$ for some $L > 0$.

Proof.

If (x,t) leaves P uniformly for all (x_0,t_0) in a uniform neighbourhood of S then by observation, (x,t) leaves P uniformly for all $(x_0,t_0) \in S$.

If (x,t) leaves P uniformly for all $(x_0,t_0) \in S$ then, by definition, there exists $\varepsilon_m, T_m > 0$

such that for all $(x_0, t_0) \in S$ then $d\big((x(t_0+t'), t_0+t'), P\big) > \varepsilon_m$ for some $0 \leq t' \leq T_m$. Now $f(x,t)$ is locally Lipschitz in Ω_x, uniformly in t, and Ω_x is compact and so $f(x,t)$ is Lipschitz in Ω_x, uniformly in t, with Lipschitz constant L. By continuous dependence on initial conditions in x [41], then for two solutions $x(t)$ and $y(t)$ of (2.1), we have $\|x(t) - y(t)\| \leq \|x_0 - y_0\| e^{L(t-t_0)}$. For $0 < b < 1$, if we set $\delta_S = b\varepsilon_m e^{-LT_m}$, $T_s = T_m$ and $\varepsilon_S = (1-b)\varepsilon_m$ then $\|x(t)-y(t)\| \leq b\varepsilon_m$ for all $\|x_0-y_0\| \leq \delta_S$ and all $t_0 \leq t \leq t_0+T_S$. Thus there exists $\varepsilon_S, \delta_S, T_S > 0$ such that for all $(x_0, t_0) \in N^x_{\delta_S}(S)$ then $d\big((x(t_0+t'), t_0+t'), P\big) > \varepsilon_S$ for some $0 \leq t' \leq T_S$. ■

We next give a set partitioning property of systems which leave sets uniformly. Lemma 5.18 gives a method to show that x leaves P uniformly in t by analysing partitioned subsets of P. The Lemma is particularly useful as it can be applied recursively.

Lemma 5.18 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$, and let $S \subset P \subset \Omega_x \times \mathbb{R}^+ \subset D$. Let there exist class $\mathcal{K}$ functions $\phi(\cdot), \sigma(\cdot)$ such that (x,t) leaves P by $\sigma\big(d_x((x_0,t_0), S)\big)$, uniformly in t, for all $(x_0,t_0) \in P \backslash S$ and (x,t) leaves S by $\phi\big(d_x((x_0,t_0), Q)\big)$, uniformly in t, for all $(x_0,t_0) \in S \backslash Q$. Then there exist class $\mathcal{K}$ function $\psi(\cdot)$ such that (x,t) leaves P by $\psi\big(d_x((x_0,t_0), Q)\big)$, uniformly in t, for all $(x_0,t_0) \in P \backslash Q$.*

Proof. For $S \backslash N^x_{\alpha_S}(Q)$ and $\alpha_S > 0$ then $\phi(\cdot)$ has an infimum $\phi(\alpha_S) > 0$. By Lemma 5.17, (x,t) leaves S uniformly for all (x_0,t_0) in a uniform neighbourhood of $S \backslash N^x_{\alpha_S}(Q)$, where we can set $\delta_S(\alpha_S) = b_S \phi(\alpha_S) e^{-LT_S}$, $\varepsilon_S(\alpha_S) = (1-b_S)\phi(\alpha_S)$. We set b_S such that $\delta_S \leq \varepsilon_S$, and note that δ_S, ε_S are class $\mathcal{K}$ functions wrt α_S. Similarly, for $P \backslash N^x_{\alpha_P}(S)$ and $\alpha_P > 0$, then (x,t) leaves P uniformly for all (x_0,t_0) in a uniform neighbourhood of $P \backslash N^x_{\alpha_P}(S)$, with constants $\delta_P = b_P \sigma(\alpha_P) e^{-LT_P}$, $\varepsilon_P = (1-b_P)\sigma(\alpha_P)$, and setting b_P such that $\delta_P \leq \varepsilon_P$. Setting $\delta_S = \alpha_P + \delta_P$ then $d_x\big((x,t), S \backslash N^x_{\alpha_S}(Q)\big) \geq \delta_S$ implies either $d_x\big((x,t), P\big) \geq \delta_P$ or $d_x\big((x,t), P \backslash N^x_{\alpha_P}(S)\big) \leq \delta_P$. Thus $d_x\big((x_0,t_0), S \backslash N^x_{\alpha_S}(Q)\big) \leq \delta_S$ implies $d_x((x,t), P) \geq \delta_P$ for some $t_0 \leq t \leq t_0 + T_P + T_S$. Similarly, setting $\alpha = \alpha_P + \alpha_S$, then $(x_0,t_0) \in P \backslash N^x_\alpha(Q)$ implies $d_x((x,t), P) \geq \delta_P$ for some $t_0 \leq t \leq t_0 + T_P + T_S$.

Setting $\kappa(\alpha_P) = \alpha_P + \delta_P = \alpha_P + b_P \sigma(\alpha_P) e^{-LT_P}$, then $\kappa(\alpha_P)$ is a class $\mathcal{K}$ function. As

$\kappa(\alpha_P) = \delta_S = b_S\phi(\alpha_S)e^{-LT_S}$ then $\alpha_P = \kappa^{-1}(b_S\phi(\alpha_S)e^{-LT_S})$ is a class $\mathcal{K}$ function wrt α_S [41]. Thus δ_P is a class $\mathcal{K}$ function wrt α_S. Now $\omega(\alpha_S) := \alpha = \kappa^{-1}(b_S\phi(\alpha_S)e^{-LT_S}) + \alpha_S$ is a class $\mathcal{K}$ function wrt α_S and so $\alpha_S = \omega^{-1}(\alpha)$ is a class $\mathcal{K}$ function wrt α [41]. Thus δ_P is a class $\mathcal{K}$ function wrt α and so there exist class $\mathcal{K}$ function $\psi(\cdot)$ such that (x,t) leaves P by $\psi\big(d_x((x_0,t_0),Q)\big)$, uniformly in t, for all $(x_0,t_0) \in P\backslash Q$. ■

5.4.2 Coordinate Transformations for Trajectories Leaving Sets

We next analyse the properties of trajectories leaving sets using coordinate transformations, in order to simplify analysis of the trajectories leaving sets.

We first give a definition of trajectories changing in the transformed coordinates. We then give a relationship between trajectories changing in the transformed coordinates and leaving sets in the original coordinates. We also introduce the basis for two methods to show that trajectories change in the transformed coordinates.

Definition 5.19 *For $y : D \to \mathbb{R}^p$, we say that $y(x,t)$ changes by $\varepsilon(x_0,t_0)$, uniformly in t, for all $(x_0,t_0) \in S$ if there exists $T > 0$ such that $\|y(x(t_0+t'),t_0+t') - y(x_0,t_0+t')\| \geq \varepsilon(x_0,t_0)$ for some $0 \leq t' \leq T$. If $p = 1$ and y decreases then we say y decreases by $\varepsilon(x_0,t_0)$, uniformly in t, for all $(x_0,t_0) \in S$.*

We next provide a relationship between trajectories leaving sets in the original coordinates and trajectories changing in the new coordinates. If we wish to study trajectories leaving a set P, then we can set new coordinates $y(x,t)$ such that $y = y(x,t) = 0$ for $(x,t) \in P$. If y becomes non-zero then x leaves set P, and so if we can determine properties of trajectories in y, then we can determine properties of trajectories in x leaving set P.

Lemma 5.20 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and $\Omega_x \times \mathbb{R}^+ \subset D$ and let there exists $Q \subset S \subset P \subset D$. Let $y : D \to \mathbb{R}^n$ such that y is locally Lipschitz, uniformly in t for all $(x,t) \in \Omega_x \times \mathbb{R}^+$ and $y(x,t) = 0$ for all $(x,t) \in P$. If there exists a class $\mathcal{K}$ function ϕ such that y changes by $\phi\big(d_x((x_0,t_0),Q)\big)$ for all $(x_0,t_0) \in S\backslash Q$ then there exists class $\mathcal{K}$ function ψ such that (x,t) leaves P by $\psi\big(d_x((x_0,t_0),Q)\big)$, uniformly*

in t, for all $(x_0, t_0) \in S \backslash Q$.

Proof. As Ω_x is compact and y is locally Lipschitz, uniformly in t, then y is Lipschitz, uniformly in t. Thus as $y(x,t) = 0$ for all $(x,t) \in P$ then $\|y(x,t)\| \leq L d_x((x,t), P)$. Thus $d_x((x,t), P) \geq \frac{1}{L}\|y(x,t)\| \geq \frac{1}{L}\phi\big(d_x((x_0,t_0), Q)\big)$ for some $t_0 \leq t \leq t_0 + T$. Therefore (x,t) leaves P by $\psi\big(d_x((x_0,t_0), Q)\big) = \frac{1}{L}\phi\big(d_x((x_0,t_0), Q)\big)$, uniformly in t, for all $(x_0, t_0) \in S \backslash Q$.

■

We next give the two methods for analysing trajectories leaving sets in a new coordinates. For the transformed coordinates $y(x,t)$, the first of the two new methods requires conditions on the direction of $\dot{y}$ but does not require bounds on the rate of change of $\dot{y}$. The second method requires the rate of change of $\dot{y}$ to be bounded but does not require conditions for the direction of $\dot{y}$.

Lemma 5.21 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and $\Omega_x \times \mathbb{R}^+ \subset D$ and let $Q_x \subset S_x \subset \Omega_x$. Let there be functions $y : \Omega_x \times \mathbb{R}^+ \to \mathbb{R}^p$ such that $\dot{y}$ is locally Lipschitz continuous, uniformly in t, for all $(x,t) \in \Omega_x \times \mathbb{R}^+$. Let there exists class $\mathcal{K}$ function $\rho(\cdot)$ and function $r : \Omega_x \to \mathbb{R}^P$ such that $\|r(x) \cdot \dot{y}(x,t)\| \geq \rho\big(d_x(x, Q_x)\big)$ and $\|r(x)\| = 1$ for all $(x,t) \in (S_x \backslash Q_x) \times \mathbb{R}^+$. For all $x \in S_x \backslash Q_x$ then let $r(x) \cdot \dot{y}(x,t)$ be sign definite for all $t \geq 0$. Then there exists a class $\mathcal{K}$ function ψ such that y changes by $\psi\big(d_x(x, Q_x)\big)$, uniformly in t, for all $(x_0, t_0) \in (S_x \backslash Q_x) \times \mathbb{R}^+$.*

Proof. We first set $y_r(x,t) = y(x,t) \cdot r(x_0)$ using $r(x_0)$ as a reference vector and so $\dot{y}_r(x,t) = \dot{y}(x,t) \cdot r(x_0)$. Now

$$\dot{y}_r(x,t) = \dot{y}_r(x_0,t) + [\dot{y}_r(x,t) - \dot{y}_r(x_0,t)] \tag{5.14}$$

Without loss of generality, we assume $\dot{y}_r(x_0,t_0) \geq 0$ and so

$$\dot{y}_r(x,t) \geq \rho\big(d_x(x_0, Q_x)\big) - L\|x - x_0\| \tag{5.15}$$

where L is the Lipschitz constant for $\dot{y}$. For neighbourhood $\|x - x_0\| \leq \frac{b}{L}\rho\big(d_x(x_0, Q_x)\big)$

where $0 < b < 1$, then $\dot{y}_r(x,t) \geq \bar{\rho}(x_0,t_0)$, where $\bar{\rho}(x_0,t_0) = (1-b)\rho\big(d_x((x_0,t_0),Q)$. As $f(x,t)$ is bounded uniformly in t then if x leaves the neighbourhood, it does so in a time greater than $\tau = \frac{b}{B_2 L}\rho\big(d_x((x_0,t_0),Q)$. Thus $y_r(x,t) \geq \bar{\rho}(x_0,t_0)t'$ for $0 \leq t' \leq \tau$ and so $y_r(x,t)$ reaches a value greater than $\tau\bar{\rho}$ in time τ. Setting $T > 0$ then $y_r(x,t)$ changes by a value greater than $\bar{\tau}\bar{\rho}$ for some $0 \leq t' \leq T$, where $\bar{\tau} = \min(\tau, T)$. As $\|y(x,t) - y(x_0,t_0)\| \geq \|y_r(x,t) - y_r(x_0,t_0)\|$ then $y(x,t)$ changes by a value greater than $\bar{\tau}\bar{\rho}$ for some $0 \leq t' \leq T$. As $\bar{\tau}\bar{\rho}$ is a class $\mathcal{K}$ function wrt $d_x(x,Q_x)$ then the result holds.

■

We next propose the second method, where $\frac{\partial \dot{y}}{\partial t}$ is bounded but there are no conditions on the direction of $\dot{y}$.

Lemma 5.22 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and let $Q \subset S \subset \Omega_x \times \mathbb{R}^+ \subset D$. Let there exist a class $\mathcal{K}$ function $\alpha(\cdot)$ and functions $y, z : D \to \mathbb{R}^p$ such that $\alpha(\|z(x,t)\|)$ is bounded on $(x,t) \in S$, and $\|\frac{\partial \dot{y}}{\partial t}\|$ is bounded and $\dot{y}$ is locally Lipschitz, uniformly in t, for all $x \in \Omega_x$ and all $t \geq 0$. Let $\|\dot{y}\| \geq \alpha(\|z(x,t)\|)$ for all $(x,t) \in S \backslash Q$. Then there exists $\kappa > 0$ such that $y(x,t)$ changes by ψ_y, uniformly in t, for all $(x_0,t_0) \in S \backslash Q$, where $\psi \geq \kappa\alpha(\|z(x_0,t_0)\|)^2$.*

Proof. For some $i = 1, \ldots, p$ then $\sqrt{p}\|\dot{y}_i(x_0,t_0)\| \geq \|\dot{y}(x_0,t_0)\|$. As y is Lipschitz and as $\|\frac{\partial \dot{y}}{\partial t}\| \leq B$ then

$$\begin{aligned}
\sqrt{p}\|\dot{y}_i(x,t)\| \geq& \sqrt{p}\|\dot{y}_i(x_0,t_0)\| - \sqrt{p}\|\dot{y}_i(x,t) - \dot{y}_i(x_0,t)\| - \sqrt{p}\|\dot{y}_i(x_0,t) - \dot{y}_i(x_0,t_0)\| \\
\geq& \|\dot{y}(x_0,t_0)\| - \sqrt{p}\|\dot{y}(x,t) - \dot{y}(x_0,t)\| - \sqrt{p}\|\dot{y}(x_0,t) - \dot{y}(x_0,t_0)\| \\
\geq& \alpha(\|z(x_0,t_0)\|) - \sqrt{p}L\|x - x_0\| - \sqrt{p}B_1\|t - t_0\| \\
\geq& \frac{1}{2}\alpha(\|z(x_0,t_0)\|) := \sqrt{p}\mu(x_0,t_0)
\end{aligned} \tag{5.16}$$

for $\|x - x_0\| \leq \frac{1}{4\sqrt{p}L}\alpha(\|z(x_0,t_0)\|)$ and $\|t - t_0\| \leq \tau_1 = \frac{1}{4\sqrt{p}B_1}\alpha(\|z(x_0,t_0)\|)$. As $\|f(x,t)\| \leq B_2$, then if x leaves the neighbourhood, it does so in a time greater than $\tau_2 = \frac{1}{4\sqrt{p}LB_2}\alpha(\|z(x,t)\|)$. Thus for $\tau = \min(\tau_1, \tau_2)$ then $\|y_i(x(t'+t_0), t'+t_0) - y_i(x_0,t_0)\| \geq \mu(x_0,t_0)t'$ for $0 \leq t' \leq \tau$ and so y changes by a value greater than $\tau\mu$ in time τ. As $\alpha(\|z\|)$ is bounded in S then

τ has a supremum T in S. Thus there exists a constant $\kappa > 0$ such that y changes by $\psi \geq \kappa\alpha(\|z(x,t)\|)^2$, uniformly in t, for all $(x_0, t_0) \in S \backslash Q$. ■

5.4.3 Lyapunov Functions and Attractivity

We next provide a lemma for attractivity in non-increasing Lyapunov functions. This lemma is similar to existing observability type results for Lyapunov functions[1–3], but adapted to be implemented with the preceding Lemmas on trajectories leaving sets uniformly.

Lemma 5.23 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and $\Omega_x \times \mathbb{R}^+ \subset D$. Let $V : D \to \mathbb{R}^+$ be a continuously differentiable function. Let there exist class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \psi(\cdot)$ such that $\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|)$, $\dot{V} \leq 0$ and V decreases by $\psi(\|x_0\|)$ for all $x_0 \in \Omega_x$ and all $t_0 \geq 0$. Then $x \to 0$ uniformly in t.*

Proof.

We first note that $V_0 = V(x_0, t_0) \leq \alpha_2(\|x_0\|)$ implies $\alpha_2^{-1}(V_0) \leq \|x_0\|$ and $\psi(\alpha_2^{-1}(V_0)) \leq \psi(\|x_0\|)$ [41]. Thus $V(x,t)$ decreases by $\psi(\alpha_2^{-1}(V_0)) := \psi_2(V_0)$, uniformly in t, where $\psi_2(\cdot)$ is a class $\mathcal{K}$ function [41].

Assume that $V \not\to 0$. Then $V \to b > 0$ as $\dot{V} \leq 0$ and V is bounded from below. Let $\mu = \psi_2(b)$ and note that there exists $T_\mu \geq 0$ such that $b \leq V(x,t) \leq b + \frac{\mu}{2}$ for all $t \geq T_\mu$. Thus $V(x,t) \leq b$ for some $t \geq 0$ as $V(x,t)$ decreases by $\psi_2(V_0) \geq \mu$, uniformly in t. This contradicts and so $V(x,t) \to 0$.

Assume that $V(x,t) \to 0$, uniformly in t, does not hold. Therefore, for some $b > 0$ and all $T_u > 0$ there exists $t_0 \geq 0$ such that $V(x(t'+t_0), t'+t_0) \geq b$ for some $t' \geq T_u$. If $b_m = \sup b$, then for all $\beta > 0$ there exists $T_\beta \geq 0$ independent of t_0 such that $V(x(t+t_0), t+t_0) \leq b_m + \beta$ for all $t \geq T_\beta$. But repeating the above argument then this contradicts, and so $V(x,t) \to 0$ uniformly in t. Now $\|x\| \leq \alpha_1^{-1}(V)$ [41] and so $x \to 0$ uniformly in t. ■

5.5 Invariance Principles for Time-Varying Systems

In this section, two invariance principles are developed and their application is shown with examples. We first provide the two main Theorems of the chapter, which introduce invariance principles for time-varying systems. The first theorem gives an invariance principle in terms trajectories leaving sets, while the second gives an invariance principle directly in terms of transformed coordinates. The first theorem has the advantage of only requiring conditions for trajectories leaving sets to hold on a subset of the domain. We then present two non-recursive implementations of the first theorem, in order to simplify the results. We also give a recursive implementation of the second theorem, which is used to show stability conditions in terms of observability matrices for linear time-varying systems.

5.5.1 Lyapunov Functions and Invariance

We next present the two invariance principles. The first theorem combines Lyapunov functions with the properties of trajectories leaving sets uniformly.

Theorem 5.24 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$, let $S \subset \Omega_x \times \mathbb{R}^+ \subset D$, and let there exist $B, \tau > 0$ such that $\|d_x((x,t),S) - d_x((x,t+t'),S)\| \leq Bt'$ for all $0 \leq t' \leq \tau$. For continuously differentiable $V : D \to \mathbb{R}^+$, let there exist class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot)$ such that $\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|)$, $\dot{V} \leq -\alpha_3(d_x((x,t),S))$ for all $x \in \Omega_x$ and all $t \geq 0$. Let there exist class $\mathcal{K}$ function $\phi(\cdot)$ such that (x,t) leave S by $\phi(\|x_0\|)$, uniformly in t, for all $(x_0,t_0) \in S\backslash\{0\} \times \mathbb{R}^+$. Then $x \to 0$, uniformly in t.*

Proof.

$$\begin{aligned} \dot{V}(x,t) &\leq -\alpha_3(d_x((x,t),S)) \\ &\leq -\alpha_3((1-2b)d_x((x_0,t_0),S)) \end{aligned} \tag{5.17}$$

for $\|x - x_0\| \leq b d_x((x_0,t_0),S)$ and $t_0 \leq t \leq t_0 + \min\left(\tau, \frac{b}{B} d_x((x_0,t_0),S)\right)$, where $0 < 2b < 1$ and $f(x,t) \leq B$. Using a similar argument to the proof of Lemma 5.21, there exists $\sigma(\cdot)$

such that V decreases by $\sigma(d_x((x_0,t_0),S))$, uniformly in t.

For $\alpha_S > 0$, from the proof of Lemma 5.18 then (x,t) leaves S uniformly for all (x_0,t_0) in a uniform neighbourhood of $S \backslash N^x_{\alpha_S}(\{0\} \times \mathbb{R}^+)$, where $\delta_S(\alpha_S)$, $\varepsilon_S(\alpha_S)$ are class $\mathcal{K}$ functions wrt α_S. We set $\delta_S(\alpha_S) \leq \varepsilon_S(\alpha_S)$ and note that $\sigma(\cdot)$ has a minimum $\sigma(\delta_S(\alpha_S))$ on $\Omega_x \times \mathbb{R}^+ \backslash N^x_{\delta_S}(S)$, which is class $\mathcal{K}$ functions wrt α_S. Setting $\alpha = \alpha_S + \delta_S$ then $\|x_0\| \geq \alpha$ implies that there exists T such that $d_x\big((x,t),S\big) \geq \delta_S$ for some $t_0 \leq t \leq t_0 + T$. Now α is class $\mathcal{K}$ functions wrt α_S and so α_S and δ_S are class $\mathcal{K}$ functions wrt α. Thus there exists a class $\mathcal{K}$ function $\psi(\cdot)$ such that V decreases by $\psi(\|x_0\|)$, uniformly in t. By Lemma 5.23 then $x \to 0$, uniformly in t. ■

We can partition the set S and then use Lemma 5.18 recursively to analyse the partitioned subsets of S independently.

The second theorem uses Lyapunov functions directly with transformed coordinates.

Theorem 5.25 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and $\Omega_x \times \mathbb{R}^+ \subset D$. Let there exists class $\mathcal{K}$ function $\gamma(\cdot)$ and functions $y_i : D \to \mathbb{R}^p$ for $i = 1, \ldots, N$ such that $\|\frac{\partial \dot{y}_i}{\partial t}\|$ is bounded and $\dot{y}$ is locally Lipschitz, uniformly in t, for all $x \in \Omega_x$ and all $t \geq 0$. Let $\|\dot{y}_i\| \geq \|y_{i+1}\|$ and let $\|y_1^T, \ldots, y_N^T\| \geq \gamma(\|x\|)$ for all $x \in \Omega_x$ and all $t \geq 0$. Let $V : D \to \mathbb{R}^+$ be a continuously differentiable function and let there exist class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot)$ such that $\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|)$, $\dot{V} \leq -\alpha_3(\|y_1\|)$ for all $x \in \Omega_x$ and $t \geq 0$. Then $x \to 0$, uniformly in t.*

Proof. $\|\dot{y}_i\| \geq \|y_{i+1}\|$ and so by Lemma 5.22, y_i changes by $\psi_i \geq \kappa_{i+1}\|y_{i+1}\|^2$, uniformly in t.

We first look at the case $(x_0,t_0) \in S_N = \{(x,t) \in D | \|y_1^T, \ldots, y_{N-1}^T\| \leq \frac{1}{2}\kappa_N \|y_N\|^2\}$. Now $\psi_{N-1} \geq \kappa_N \|y_N\|^2$ and so for all $(x_0,t_0) \in S_N$ then there exists $T_a \geq 0$ such that $\|y_{N-1}(x(t_a),t_a)\| \geq \frac{1}{2}\kappa_N \|y_N(x_0,t_0)\|^2$ for some $t_0 \leq t_a \leq t_0 + T_a$. It can be noted that $\frac{1}{2}\kappa_N(\cdot)^2$ is a class $\mathcal{K}$ function. Thus by recursion, then there exists some $T_N \geq 0$ and class $\mathcal{K}$ function $\phi(\cdot)$ such that $\|y_1(x(t'),t')\| \geq \phi(\|y_N(x_0,t_0)\|)$ for some $t_0 \leq t' \leq t_0 + T_N$ and all $(x_0,t_0) \in S_N$. For $\|y_1^T, \ldots, y_{N-1}^T\| \leq \frac{1}{2}\kappa_N \|y_N\|^2$ then $(\frac{1}{2}\kappa_N \|y_N\|^2)^2 + \|y_N\|^2 \geq$

$\|y_1^T, \ldots, y_{N-1}^T\|^2 + \|y_N\|^2 \geq \gamma(\|x\|)^2$. Setting $\rho(\cdot) = \frac{1}{4}\kappa_N^2(\cdot)^4 + (\cdot)^2$, then $\rho(\cdot)$ is a class $\mathcal{K}$ function and so $\phi(\rho^{-1}(\gamma(\cdot)^2))$ is a class $\mathcal{K}$ function [41]. Thus there exists a class $\mathcal{K}$ function β_N such that $\|y_1(x(t'), t')\| \geq \beta_N(\|x_0\|)$ for some $t_0 \leq t' \leq t_0 + T_N$ and all $(x_0, t_0) \in S_N$.

For $w := \|y_1^T, \ldots, y_{N-1}^T\| \geq \frac{1}{2}\kappa_N\|y_N\|^2$, then $w^2 + \frac{2}{\kappa_N}w \geq \|y_1^T, \ldots, y_{N-1}^T\|^2 + \|y_N\|^2 \geq \gamma(\|x\|)^2$. Thus $w \geq \sqrt{\left(\frac{2}{\kappa}\right)^2 + 4\gamma(\|x\|)^2} - \frac{2}{\kappa}$, where the RHS is a class $\mathcal{K}$ function. Thus $\|y_1^T, \ldots, y_{N-1}^T\| \geq \gamma_{N-1}(\|x\|)$ for some class $\mathcal{K}$ function $\gamma_{N-1}(\cdot)$.

Repeating the argument, we can find $T_i > 0$ and class $\mathcal{K}$ functions $\beta_i(\cdot)$ such that $\|y_1(x(t'), t')\| \geq \beta_i(\|x_0\|)$ for some $t_0 \leq t' \leq t_0 + T_i$ and all $(x_0, t_0) \in S_i = \{(x, t) \in D | \|y_1^T, \ldots, y_{i-1}^T\| \leq \frac{1}{2}\kappa_i\|y_i\|^2, \|y_1^T, \ldots, y_i^T\| \geq \frac{1}{2}\kappa_{i+1}\|y_{i+1}\|^2\}$ for $i = 1, \ldots, N$. Setting $\bar{\beta} = \min(\beta_1(\cdot), \ldots, \beta_1(\cdot))$ and $\bar{T} = \min(T_1, \ldots, T_N)$, then $\|y_1(x(t'), t')\| \geq \bar{\beta}(\|x_0\|)$ for some $t_0 \leq t' \leq t_0 + \bar{T}$ and all $(x_0, t_0) \in \Omega_x \times \mathbb{R}^+$. Using a similar argument to the proof of Theorem 5.24, then $x \to 0$, uniformly in t. ■

5.5.2 Non-Recursive Methods

We give two methods to implement Theorem 5.24 non-recursively using transformed coordinates, which allows simplified implementation of the Theorem. We analyse a pendulum with time-varying parameters to show the application of the methods.

The first method requires conditions on the direction of $\dot{y}$ of the transformed coordinates $y(x, t)$, but does not require bounds on the rate of change of $\dot{y}$.

Corollary 5.26 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$, let $\Omega_x \times \mathbb{R}^+ \subset D$ and let $S_x \subset \Omega_x$. For continuously differentiable $V : D \to \mathbb{R}^+$, let there exist a continuous function $W(x)$ and class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot)$ such that $\alpha_1(\|x\|) \leq V(x, t) \leq \alpha_2(\|x\|)$, $\dot{V} \leq -W(x)$ for all $x \in \Omega_x$ and all $t \geq 0$. Let there be functions $y : \Omega_x \times \mathbb{R}^+ \to \mathbb{R}^p$ such that $y(x, t) = 0$ for all $W(x) = 0$ and let y and $\dot{y}$ be locally Lipschitz continuous, uniformly in t, for all $(x, t) \in \Omega_x \times \mathbb{R}^+$. Let there exists continuous function $\mu(\cdot)$ which is positive definite for all $x \in S_x \backslash \{0\}$ and function $r : D_x \to \mathbb{R}^P$ such that $\|r(x)\| = 1$ for all*

$x \in S_x \backslash \{0\}$. *Let there exists* $\|r(x) \cdot \dot{y}(x,t)\| \geq \mu(x)$ *for all* $(x,t) \in (S_x \times \mathbb{R}^+) \backslash (\{0\} \times \mathbb{R}^+)$ *and for each* x *let* $r(x) \cdot \dot{y}(x,t)$ *be sign definite for all* $t \geq 0$. *Then* $x \to 0$ *uniformly in* t.

Proof. Setting $Q_x = \{0\}$ and $S_x = \{x \in \Omega_x | W(x) = 0\}$, then there exists class $\mathcal{K}$ function $\rho(\cdot)$ such that $\mu(x) > \rho(\|x\|)$ [41] and so $\|r(x) \cdot \dot{y}(x,t)\| \geq \rho(\|x\|)$ for all $(x,t) \in S \backslash (Q_x \times \mathbb{R}^+)$. By Lemma 5.21 there exists $\mathcal{K}$ function $\psi_y(\cdot)$ such that y changes by $\psi_y(\|x\|)$, uniformly in t, for all $(x_0, t_0) \in S \backslash (Q_x \times \mathbb{R}^+)$. By Lemma 5.20 then there exists $\mathcal{K}$ function $\phi(\cdot)$ such that(x,t) leaves S by $\phi(\|x\|)$, uniformly in t, for all $(x_0, t_0) \in S \backslash (Q_x \times \mathbb{R}^+)$

Now there also exists a class $\mathcal{K}$ function $\alpha_3(\cdot)$ such that $\alpha_3(d_x(x, S_x)) \leq W(x)$ [41] and so $\dot{V} \leq -W(x) \leq -\alpha_3(d_x(x, S_x))$. By Theorem 5.24 then $x \to 0$ uniformly in t. ■

We next apply the first method to the analysis of the stability of a pendulum with time-varying parameters.

Example 5.27 *Consider*

$$\begin{aligned} \dot{x}_1 &= -\gamma(t) x_1 - \beta(t) \sin x_2 \\ \dot{x}_2 &= x_1 \end{aligned} \tag{5.18}$$

where $0 < \beta_l \leq \beta(t) \leq \beta_h$ *is a continuous function,* $\gamma(t)$ *is a bounded continuous function and* $2\gamma(t)\beta(t) + \dot{\beta}(t) \geq p > 0$. *Let* $\Omega_x = \{x \in \mathbb{R}^2 | \frac{1}{2\beta_h} x_1^2 + (1 - \cos x_2) \leq 2, |x_2| \leq \pi\}$, $x_0 \in \{x \in \mathbb{R}^2 | \frac{1}{2\beta_l} x_1^2 + (1 - \cos x_2) \leq c, |x_2| \leq \pi\}$, *where* $c < 2$. *We use Corollary 5.26 with*

$$V = \frac{1}{2\beta(t)} x_1^2 + (1 - \cos x_2) \tag{5.19}$$

$$\dot{V} = -\frac{1}{2\beta(t)^2} (2\gamma(t)\beta(t) + \dot{\beta}(t)) x_1^2 \leq -\frac{p}{2\beta_h^2} x_1^2 \leq 0 \tag{5.20}$$

Setting $y = x_1$ *and* $r = 1$ *then for* $y = x_1 = 0$ *we have* $\dot{y} = -\beta(t) \sin x_2$, *where* $|\dot{y}| \geq \beta_l |\sin x_2| = \mu(x) > 0$ *for* $x_2 \neq 0$. *Thus for system* (5.18), $(x_1, x_2) \to 0$ *uniformly in* t.

Physically, the Lyapunov function used is the mechanical energy of a pendulum normalised by $\beta(t)$, a time-varying parameter. It is interesting to observe that $\gamma(t) \geq 0$ does not necessarily have to hold, depending upon $\dot{\beta}(t)$.

We next present the second method to implement Theorem 2.15 non-recursively. The second method requires the rate of change of $\dot{y}$ of the transformed coordinates to be bounded but does not require conditions for the direction of $\dot{y}$.

Corollary 5.28 *Let Ω_x be a compact set where $x(t) \in \Omega_x$ for all $t \geq t_0$ and $\Omega_x \times \mathbb{R}^+ \subset D$. For continuously differentiable $V : D \to \mathbb{R}^+$, let there exist class $\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot)$ and function $y : D \to \mathbb{R}^p$ such that $\alpha_1(\|x\|) \leq V(x,t) \leq \alpha_2(\|x\|)$, $\dot{V} \leq -W(x)$. Let $y(x,t)$ be locally Lipschitz, uniformly in t and let $\|\frac{\partial y}{\partial t}\|$ be bounded. Let $S = \{(x,t) \in \Omega_x \times \mathbb{R}^+ | W(x) = 0\}$ and let $Q = \{0\} \times \mathbb{R}^+$. If $y(x,t) = 0$ for all $(x,t) \in S$ and $\|\dot{y}(x,t)\| \geq \mu(x)$ for all $x \in S \backslash Q$, then $x \to 0$ uniformly in t.*

Proof. The proof is similar to the proof of Corollary 5.26, with Lemma 5.22 used instead of Lemma 5.21. ■

Example 5.29 *We can apply Corollary 5.28 to Equation 5.18 in Example 5.27. This can also show that $x \to 0$ uniformly if we have the extra conditions that $\gamma'(t)$ and $\dot{\beta}(t)$ are bounded.*

5.5.3 Output Recursive Methods and Observability

We next show the methodology of Theorem 5.25 for a recursive application by deriving stability condition based on observability matrices for time-varying systems. As well as testing a recursive method, this also show the connection with observability for linear time-varying systems [61, 77].

Corollary 5.30 *Let there exist continuously differentiable $P : \mathbb{R}^+ \to \mathbb{R}$ such that $k_1 I \preceq P(t) \preceq k_2 I$ and*

$$P(t)A(t) + \dot{P}(t) + A^T(t)P(t) \preceq -C(t)^T C(t) \tag{5.21}$$

Let $L_1 = C(t)$ and let $L_{i+1}^T L_{i+1} \preceq (\dot{L}_i + L_i A)^T (\dot{L}_i + L_i A)$ for $2 = 1, \ldots, m$. For $L = [L_1^T, \ldots, L_m^T]^T$, if there exist $k > 0$ such that $\dot{L}$ is bounded and $L^T L \succeq k I_n$ then $x \to 0$ uniformly in t.

Proof. Setting $V = x^T P x$ and $y_i = L_i x$ for $i = 1, \ldots, m-1$, then $\dot{V} \leq -y_1^T y_1$, $\|\dot{y}_i\| \geq \|y_{i+1}\|$ for $i = 1 \ldots, m-1$. By Theorem 5.25 then $x \to 0$, uniformly in t. ■

Choosing $L_{i+1}(t) = \dot{L}_i(t) + L_i(t)A(t)$ is possible if $A(t)$ and $C(t)$ are sufficiently smooth. Other choices of L_i are required if this condition does not hold, such as setting L_i as constant matrices.

For an LTI system with $m = N$, $L_1 = C$ and $L_{i+1} = L_i A = CA^i$, then $L^T L \geq kI_N$ is equivalent to the observability matrix having a rank of N, and so the linear time-varying and linear time-invariant cases are equivalent.

5.5.4 Comparison to Existing Methods

We next compare the presented invariance principle in Theorem 2.15 and its corollaries to Matrosov's theorem, observability theorems and limiting equation methods.

Theorem 2.15 and its corollaries are similar to the original version of Matrosov's Theorem if we treat y as a scalar auxiliary function, but the conditions in Corollaries 5.26 and 5.28 for the derivative of auxiliary function y are only required for $E = \{x \in \Omega_x | W(x) = 0\}$, rather than variously defined neighbourhoods of E as in Matrosov's original theorem [59], [74]. This simplifies analysis and allows a treatment more consistent with Krasovskii-Lasalle's Invariance principle and observability conditions. In Example 5.27, Matrosov's original Theorem requires analysis of $\dot{y} = -\gamma(t)x_1 - \beta(t)\sin x_2$ rather than $\dot{y} = -\beta(t)\sin x_2$ and requires analysis for small values of x_1 rather than $x_1 = 0$ [59], [74].

Theorem 2.15 and its corollaries differ from some more recent generalisations of Matrosov's Theorem (See [57] and references therein) requiring the derivative of the auxiliary function y to be sign definite, which is a more restrictive condition and does not allow comparison with observability conditions. Similarly, more recent generalisations require a triangular output, where the first output is a variable of 1 state, the second 2 states and so on, which restricts applications. For Example 5.27, $y = x_1$ cannot be used with existing methods and so the function $y = x_1 x_2$ is used instead. The existing method is less intuitive, more

difficult to find and analyse y, and is more difficult to compare to other results in systems theory, such as observability.

Theorem 2.15 and its corollaries are simpler than using limiting equation or differential inclusion techniques as they apply to the vector field directly. Also, differential inclusion methods do not typically allow for a time-varying Lyapunov function as we used in the method above. The method in Example 5.27 is simpler than observability-type methods, where we treat y as an output function, as we do not need to confirm conditions along system trajectories.

Theorem 2.15 and its corollaries are simpler than observability-type methods, where we treat y as an output function, as we do not need to confirm conditions along system trajectories.

5.6 Applications to Nonlinear, Large-Scale, Time-Varying Systems

In this section we show an application of the proposed technique for the case of networked pendulums with time-varying parameters.

Example 5.31 *We apply Corollary 5.28 to the system*

$$\begin{aligned} \dot{x}_1 &= -\Gamma(t)x_1 - B(t)\sin x_2 \\ \dot{x}_2 &= x_1 \end{aligned} \tag{5.22}$$

where $x_1, x_2 \in \mathbb{R}^N, B, \Gamma : \mathbb{R}^+ \to \mathbb{R}^{N\times N}$, $\sin x_2 = (\sin x_{21}, \ldots, \sin x_{2N})^T$, $B(t)$ *is a continuous, symmetric function with bounded* $\dot{B}(t)$ *and* $0 \prec \beta_l I_N \preceq B(t) \preceq \beta_h I_N$, $\Gamma(t)$ *is a bounded continuous function and* $\Gamma(t)B(t) + B(t)\Gamma(t)^T + \dot{B} \succeq pI_N \succ 0$. *Let* $\Omega_x = \{x \in \mathbb{R}^2 | \frac{1}{2\beta_h} x_1^T x_1 + \sum_i (1 - \cos x_{2i}) \leq 2, |x_{2i}| \leq \pi\}$, $x_0 \in \{x \in \mathbb{R}^2 | \frac{1}{2\beta_l} x_1^T x_1 + \sum_i (1 - \cos x_{2i}) \leq c, |x_{2i}| \leq \pi\}$, *where* $c < 2$. *We first note that* $0 \prec \frac{1}{\beta_h} I_N \preceq B(t)^{-1} \preceq \frac{1}{\beta_l} I_N$ *from above and*

$B^TB \succeq \beta_l^2 I_N$, $(B^{-1})^T B^{-1} \succeq \frac{1}{\beta_h^2} I_N$ *as* B *is symmetric. Now*

$$\begin{aligned} V &= \frac{1}{2} x_1^T B^{-1}(t) x_1 + \sum_i (1 - \cos x_{2i}) \\ \dot{V} &= -\frac{1}{2} x_1^T (B^{-1}\Gamma + \Gamma^T B^{-1} + B^{-1}\dot{B}B^{-1}) x_1 \\ &= -\frac{1}{2} x_1^T (B^{-1})^T (\Gamma B + B\Gamma^T + \dot{B}) B^{-1} x_1 \leq -\frac{p}{2\beta_h^2} x_1^T x_1 \leq 0 \end{aligned} \tag{5.23}$$

Using $y = x_1$ *then for* $y = x_1 = 0$ *we have* $\dot{y} = -B(t)\sin x_2$. *Now* $\|\dot{y}\| = \|B(t)\sin x_2\| = \sqrt{\sin x_2^T B^T B \sin x_2} \geq \beta_l \|\sin x_2\| = \mu(\|x\|) > 0$ *for* $x_2 \neq 0$. *Thus* $(x_1, x_2) \to 0$ *uniformly in* t *by Corollary 5.28.*

This example is an interesting application as we do not require that $B(t)$ and $\Gamma(t)$ to be (upper) triangular, as required by standard Matrosov's Theorem results [57].

Example 5.32 *We can use Corollary 5.26 for Example 5.31 and remove the assumption that* $\dot{B}(t)$ *is bounded. Using the same* V, $\dot{V}$ *and* y *as for Example 5.31 then for* $y = x_1 = 0$ *we have* $\dot{y} = -B(t)\sin x_2$. *Now setting* $r(x) = \frac{\sin(x_2)}{\|\sin(x_2)\|}$ *then* $\|r(x)\cdot\dot{y}\| = \frac{1}{\|\sin x_2\|}\|(\sin x_2)^T B(t)\sin x_2\| \geq \beta_l \|\sin x_2\| > 0$ *for* $x_2 \neq 0$. *Thus* $(x_1, x_2) \to 0$ *uniformly in* t *by Corollary 5.26.*

Example 5.32 show Corollary 5.26 gives a more general condition than Corollary 5.28 at the tradeoff of having to find $r(x)$.

5.7 Conclusion

In this chapter we have provided a novel proof of the Krasovskii-Lasalle Invariance principle for forward time. This was achieved using the invariance properties of the region of attraction, rather than the invariance property of the limit set. We then proposed and proved invariance principles for time-varying systems, where we analyse trajectories leaving sets. This can be viewed as a time-varying equivalent to Barbashin-Krasovskii Theorem [15, 45], by introducing trajectories leaving sets 'uniformly in t'. We also pro-

posed methods to analyse trajectories leaving sets using transformed coordinates, which simplifies analysis. Future work will further apply the methodology, particularly for recursive results for non-autonomous stability.

Chapter 6

Conclusion

In this thesis, new methods have been developed for analysing nonlinear systems with multiple challenges. Both analytical and computational methods have been developed to analyse nonlinear systems, which are either large-scale, non-polynomial, time-varying or uncertain. These nonlinear systems with multiple challenges present both interesting and challenging theoretical questions, as well as being related to important applications.

In Chapter 3, Generalised Absolute Stability theory was developed, which gives a framework to analyse nonlinear systems which are non-polynomial, time-varying or uncertain using sum of squares programming. This work extends existing Absolute Stability theory, reducing conservatism for when system states are 'further away' from the equilibrium. This work also extends the applicability of Sum of Squares programming to analysing systems beyond polynomial systems.

In Chapter 4, Perturbation methods for invariant Sum of Squares (SOS) and Semidefinite Programs (SDP) were introduced, which can significantly improve scalability of SDPs, allowing sum of squares programming to be used to analyse large-scale nonlinear systems. Methods for invariant or symmetric SOS and SDPs give significant computational benefit for scalability and other numerical issues. However, these methods can currently only be used for a small class of problems, limiting applicability. By introducing perturbation

methods, the advantages of analysing invariant or symmetric SOS/SDPs can be extended to a much larger class of problems. In the examples in Chapter 4, this allowed sum of squares problems of more than 100 variables to be analysed, where existing methods could only treat problems of 10-15 variables. When used appropriately, the 'gap' introduced in the SDP feasibility was found to be small.

In Chapter 5, new invariance principles were introduced which give an analytical method for analysing nonlinear, time-varying systems. Currently, a complete invariance principle only exists for autonomous systems. In this thesis, a new proof of the existing principle was presented, giving a different perspective to the problem. Invariance principles were then introduced for the non-autonomous case, which simplify, unify and generalise a number of different existing methods for invariance in non-autonomous systems. When applied to examples, this work greatly extends the applicability and practical use over existing theorems.

Common Themes

A primary theme across the thesis is the analysis of nonlinear systems with multiple challenging properties, such as nonlinear systems which are also large scale, time-varying, non-polynomial or uncertain. Although nonlinear systems theory has been developed over a long period of time, further work is required for nonlinearity together with these challenges. Chapters 3 and 5 give different methods for studying time-varying, nonlinear systems, Chapters 4 and 5 give methods applicable to large-scale, nonlinear systems and Chapter 3 gives methods to analyse uncertain or non-polynomial systems.

A second theme across the thesis is developing and applying methods for SOS programming. Generally, further work is required on SOS programming in three main areas of scalability, new frameworks and non convex SOS problems, of which the first two are treated in this thesis. Chapter 3 proposes a new framework of generalised absolute stability for time-varying, uncertain or non-polynomial systems. This is important as it

both improves existing absolute stability techniques for nonlinear systems as well as extending the applicability of SOS programming. In contrast, Chapter 4 extends the SOS programming methodology to improve the scalability of SOS. This is important as the current major limitation of using SOS programming is that it can typically only be used for systems with 10-15 variables [30].

Finally, all chapters are motivated to develop methods for finding Lyapunov functions. Since finding a Lyapunov function is the primary challenge in the Lyapunov theory, simplifying these methods is important in both better understanding existing problems and analysing new problems. In Chapter 5, the methods to find Lyapunov functions are analytical while in Chapters 3 and 4, the developed methodology uses computational approaches.

Future Directions

Interesting future directions of research for generalised absolute stability exist in applications as well as developing bounds for feedback with memory. Immediate applications for generalised absolute stability involve analysis of systems with non-polynomial or uncertain functions, such as power systems and biological systems. The methodology was partly motivated by analysing uncertainty in biological systems. Biological systems are typically described by models with rational functions (e.g. gene expression), and so the generalised sector has the potential to describe uncertainty in biology systems while retaining the rational function structure. The pendulum example can be viewed as a proof of concept for applications of the method to power systems. Another partial motivation for the technique was for bounding higher order terms and errors in model reduction, allowing sum of squares to be used for larger systems. A future application is to use the method together with model reduction. Another future direction is to develop generalised absolute stability with integral polynomial constraints, which would more closely tie the developed theory to the existing work on integral quadratic constraints [60]. This would allow feed-

back terms with memory, rather than the memoryless feedback terms used in generalised absolute stability. Applications of this would be in analysing uncertain systems and in analysing models which have been reduced.

Work is required on perturbation methods of invariant SOS and SDPs to decrease the conservatism in results for a wider range of cases, rather than improving scalability, as was discussed extensively in Section 4.7. This future work will focus on extending the applications of the current methodology, developing new methods to decompose any SDP or SOS problem polynomial into a convex combination of near invariant matrices or polynomials, and methods to find appropriate representations to use as inputs to the Algorithms. The primary focus of future work will be on developing methods in Chapter 4, as this has the most research potential.

Future directions of research for the invariance principle are extending the recursive methodology, extending the theory beyond non-autonomous systems, such as partial stability, as well as further applications. The recursive methodology for trajectories leaving sets uniformly requires testing and applications. Also, the methodology can be extended beyond non-autonomous systems to partial stability and set stability. Finally, all developed methods require further use in applications.

Bibliography

[1] D. Aeyels. Asymptotic stability of nonautonomous systems by Liapunov's direct method. *Systems and Control Letters*, 25(4):273–280, 1995.

[2] D. Aeyels and J. Peuteman. A new asymptotic stability criterion for nonlinear time-variant differential equations. *IEEE Transactions on Automatic Control*, 43(7):968–971, 1998.

[3] D. Aeyels, R. Sepulchre, and J. Peuteman. Asymptotic stability for time-variant systems and observability: Uniform and nonuniform criteria. *Mathematics of Control, Signals and Systems*, 11(1):1–27, 1998.

[4] M.J. Allen, J. Ryan, C.E. Hanson, and J.F. Parle. String stability of a linear formation flight control system. *Nasa Technical Report*, 2002.

[5] U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. CRC Press, 2007.

[6] J. Anderson and A. Papachristodoulou. Robust nonlinear stability and performance analysis of an F/A-18 aircraft model using sum of squares programming. *International Journal of Robust and Nonlinear Control*, 2012.

[7] P. Apkarian and H.D. Tuan. Parameterized LMIs in control theory. *SIAM Journal on Control and Optimization*, 38:1241–1264, 2000.

[8] M. Artin. *Algebra*. Pearson Education, 2011.

[9] Z. Artstein. *Limiting Equations and Stability of Nonautonomous Ordinary Differen-*

tial Equations (Appendix A of 'The Stability of Dynamical Systems', J.P. Lasalle). SIAM Regional Conference Series in Applied Mathematics, 1976.

[10] Z. Artstein. The limiting equations of nonautonomous ordinary differential equations. *Journal of Differential Equations*, 25:184–202, 1977.

[11] Z. Artstein. Uniform asymptotic stability via the limiting equations. *Journal of Differential Equations*, 27:172–189, 1978.

[12] Z. Artstein. Stability, observability and invariance. *Journal of Differential Equations*, 44(2):224–248, 1982.

[13] G. Baker and J. Blackburn. *The Pendulum: A Case Study in Physics.* OUP Oxford, 2005.

[14] I. Barbalat. Systemes d'equations differentielles d'oscillations non lineaires. *Rev. Roumaine Math. Pures Appl.*, 4(2):267–270, 1959.

[15] E. Barbashin and N. Krasovskii. On the stability of motion in the large (in russian). *Dokl. Akad. Nauk. USSR*, 86:453–456, 1952.

[16] A.R. Bergen and D.J. Hill. A structure preserving model for power system analysis. *IEEE Transactions on Power Apparatus and Systems*, 100(1):25–35, 1981.

[17] A.R. Bergen and V. Vittal. *Power Systems Analysis.* Prentice Hall, 2000.

[18] N.P. Bhatia and G.P. Szegő. *Stability Theory of Dynamical Systems.* Springer-Verlag, 2000.

[19] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang. *Complex Network: Structure and Dynamics.* Elsevier, Physics Reports, 2006.

[20] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in Systems and Control.* SIAM, 1994.

[21] S. Boyd and L.Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[22] B.Reznick. Extremal psd forms with few terms. *Duke Mathematical Journal*, 45(2):363–374, 1978.

[23] G. Chesi. Domain of attraction: Estimates for non-polynomial systems via LMIs. *Proc. 16th IFAC World Congress on Automatic Control*, 2005.

[24] G. Chesi. Estimating the domain of attraction for non-polynomial systems via LMI optimizations. *Automatica*, 45:15361541, 2009.

[25] G. Chesi. LMI techniques for optimization over polynomials in control: a survey. *IEEE Transactions in Automatic Control*, 55(11):2500–2510, 2010.

[26] M.D. Choi, T.Y. Lam, and B. Reznick. Sums of squares of real polynomials. *Proceedings of Symposia in Pure Mathematics*, 58(2):103–126, 1995.

[27] K. Gaterman and P.A. Parrilo. Symmetry groups, semidefinite programs, and sum of squares. *Journal of Pure and Applied Algebra*, 192:95–128, 2004.

[28] Ji-guang Sun Gilbert W. Stewart. *Matrix perturbation theory*. Academic Press, 1990.

[29] W.M. Haddad and V. Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov Approach*. Princeton University Press, 2008.

[30] E.J. Hancock and A. Papachristodoulou. Structured sum of squares. *Proceedings of the IEEE CDC*, 2011.

[31] E.J. Hancock and A. Papachristodoulou. An invariance principle for time-varying systems. *Proceedings of the IEEE CDC*, 2012.

[32] E.J. Hancock and A. Papachristodoulou. Generalised absolute stability and sum of squares. *Automatica*, 49(4):960–967, 2013.

[33] J.L. Hemmen and W.F. Wreszinski. Lyapunov functions for the Kuramoto model of nonlinearly coupled oscillators. *Journal of Statistical Physics*, 72(1/2):145–166, 1993.

[34] H. Hindi and S. Boyd. Analysis of linear systems with saturation using convex optimisation. *Proceedings on the 37th IEEE CDC*, 1998.

[35] G.H. Hines, M. Arcak, and A.K. Packard. Equilibrium-independent passivity: A new definition and numerical certification. *Automatica*, 47:1949–1956, 2011.

[36] R.A. Horn and C.A. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[37] S.B. Hsu. A survey of constructing Lyapunov functions for mathematical models in populations biology. *Taiwanese Journal of Mathematics*, 9(2):151–173, 2005.

[38] T. Hu, B. Huang, and Z. Lin. Absolute stability with a generalized sector condition. *IEEE Transactions on Automatic Control*, 49(4), 2004.

[39] T. Hua, Z. Lin, and B. Chen. An analysis and design method for linear systems subject to actuator saturation and disturbance. *Automatica*, 38:351359, 2002.

[40] A. Jadbabaie, N. Motee, and M. Barahona. On the stability of the Kuramoto model of coupled nonlinear oscillators. *Proceedings of the 2004 American Control Conference*, 2004.

[41] H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[42] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, and R. Herwig. *Systems Biology*. WILEY-VCH, 2009.

[43] M. Kojima, S. Kim, and H. Waki. Sparsity in sum of squares polynomials. *Mathematical Programming*, 103(1):45–62, 2005.

[44] E. Kosmatopoulos and M. Christodoulou. Structural properties of gradient recurrent high-order neural networks. *IEEE Transactions on Circuits and Systems- II*, 42(9), 1995.

[45] N. Krasovskii. Problems of the theory of stability of motion. *Stanford University Press*, 1963.

[46] J. Lasalle. The extent of asymptotic stability. *Proceedings of the National Academy of Sciences*, 46:363–365, 1960.

[47] J. Lasalle. Some extensions of Liapunov's second method. *IRE Trans. Circuit Theory*, 7(4):520–527, 1960.

[48] J. Lasalle. Stability theory for ordinary differential equations. *Journal of Differential Equations*, 4:57–65, 1968.

[49] T. Lee and Z. Jiang. A generalization of Krasovskii-lasalle theorem for nonlinear time-varying systems: Converse results and applications. *IEEE Transactions on Automatic Control*, 50(8):1147–1163, 2005.

[50] T. Lee, D. Liaw, and B. Chen. A general invariance principle for nonlinear time-varying systems and its applications. *IEEE Transactions on Automatic Control*, 46(12):1989–1993, 2001.

[51] X. Liao and P. Yu. *Absolute Stability of Nonlinear Control Systems*. Springer Science, New York, 2008.

[52] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[53] J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011, 2009.

[54] H. Logemann and E.P. Ryan. Non-autonomous systems: Asymptotic behaviour and weak invariance principles. *Journal of Differential Equations*, 189:440–460, 2003.

[55] H. Logemann and E.P. Ryan. Asymptotic behaviour of nonlinear systems. *The Mathematical Association of America*, 111:864–889, 2005.

[56] W. Lohmiller and J.J.E. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6), 1998.

[57] A. Loria, E. Panteley, D. Popovic, and A. Teel. A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems. *IEEE Transactions on Automatic Control*, 50(2):183–198, 2005.

[58] J. Machowski, J.W. Bailek, and J.R. Bumby. *Power System Stability and Control*. Wiley, 2008.

[59] V.M. Matrosov. On the stability of motion. *Prikl. Mat. MNleh.*, 26:885–895, 1962.

[60] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6), 1997.

[61] R.H. Middleton and G.C. Goodwin. Adaptive control of time-varying linear systems. *IEEE Transactions on Automatic Control*, 33(2):150–155, 1988.

[62] J.D. Murray. *Mathematical Biology I: An Introduction*. Springer, 2002.

[63] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[64] J. Nie. Sum of squares method for sensor network localisation. *Computational Optimization and Applications*, 43(2):151–179, 2009.

[65] A. Papachristodoulou. Scalable analysis of nonlinear systems using convex optimization. *Ph.D. dissertation, Caltech*, 2005.

[66] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using sum of squares decomposition. *Proceedings of the IEEE CDC*, 2002.

[67] A. Papachristodoulou and S. Prajna. Analysis of non-polynomial systems using the sum of squares decomposition. *Positive Polynomials in Control*, 312:23–43, 2005.

[68] P.A. Parrilo. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. *PhD Thesis*, 2000.

[69] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96:293–320, 2003.

[70] C. Pittet, S. Tarbouriech, and C. Burgat. Stability regions for linear systems with saturating controls via circle and Popov criteria. *Proceedings of the 36th IEEE conference on decision and control, San Diego, California*, pages 4518–4523, 1997.

[71] S. Prajna, A. Papachristodoulou, and P.A. Parrilo. Introducing SOSTOOLS: A general purpose sum of squares program solver. *Proceedings of the IEEE CDC*, 2002.

[72] Y. Pykh. Lyapunov functions for lotka-volterra systems: An overview and problems. *Proceedings of 5th IFAC Symposium "Nonlinear Control Systems"*, 2001.

[73] K.C. Toh R.H Tutuncu and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming Ser. B*, 95:189–217, 2003.

[74] N. Rouche, P. Habets, and M. Laloy. *Stability Theory by Liapunov's Direct Method*. Springer, 1977.

[75] J.P. Serre. *Linear Representations of Finite Groups,*. Springer, Berlin, 1977.

[76] R. Shone. *Economic Dynamics: Phase Diagrams and their Economic Applications*. Cambridge University Press, 2002.

[77] L.M. Silverman and H.E. Meadows. Controllability and observability in time-variable linear systems. *SIAM Journal on Control*, 5(1):64–73, 1967.

[78] J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[79] S. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Pres, 1994.

[80] J.F. Sturm. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, pages 625–653, 1999.

[81] J.A.K. Suykens, J. Vandewalle, and B. De Moor. An absolute stability criterion for the Lure problem with sector and slope restricted nonlinearities. *IEEE Transactions on Circuits and Systems*, 45(9):1007–1009, 1998.

[82] W. Tan and A. Packard. Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. *IEEE Transactions On Automatic Control*, 53(2):565–571, 2008.

[83] B. Tibken and Y. Fan. Computing the domain of attraction for polynomial systems

via BMI optimization method. *Proceedings of the American Control Conference*, 2006.

[84] U. Topcu and A. Packard. Local stability analysis for uncertain nonlinear systems. *Automatic Control, IEEE Transactions on*, 54(5):1042–1047, 2009.

[85] U. Topcu, A. Packard, and P. Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44:2669–2675, 2008.

[86] F. Vallentin. Symmetry in semidefinite programs. *Linear Algebra and its Applications*, 430:360–369, 2009.

[87] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimisation problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.

[88] X.F. Wang and G. Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6 – 20, 2003.

[89] T. Yoshizawa. Asymptotic behaviour of solutions of a systems of differential equations. *Cont. Diff. Eqs.*, 1:371–388, 1963.