Dynamical System Decomposition and Analysis Using Convex Optimization

James Anderson

Department of Engineering Science

University of Oxford
Christ Church

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy at the University of Oxford

Hilary Term, 2012

To Mum, Dad, Alex and Helen

Acknowledgements

Given that this section is the least technical part of the thesis by assumption it should have been the easiest to write. However this is not the case. There are many people in the department, control group, DTC and college who over the last few years have made my time in Oxford a truly unique and enjoyable experience. Finding words to sufficiently express my gratitude to everyone is not an easy task.

First and foremost I would like to thank my supervisor Antonis Papachristodoulou for his continual support and guidance - both personal and professional. I am especially grateful for the freedom he gave me to find interesting problems to work on and for the countless discussions we have had over the years. His ability to see the big picture while keeping track of the technical details is something I will continue to strive for. Staying in the control group, I would also like to thank Stephen Duncan for many insightful discussions related to this work. A special thanks go to Steve for trusting me to try my hand at undergraduate teaching while he was on sabbatical. I have thoroughly enjoyed my time in the group and thank everyone, past and present, for the good times. Specifically thanks to Johannes Buerger and Ed Hancock for the not insignificant number of coffee breaks we've taken together and the conversations about everything from football to politics that followed, I think once we even talked about control. Additionally thanks are due to André Teixeira and Henrik Sandberg for the collaboration that resulted in the work that appears in the beginning of Chapter 6.

None of this work would have been possible if it wasn't for David Gavaghan and the University of Oxford's Doctoral Training Centre. During my year at the DTC I made good friends and learnt a lot, the most notable lesson being that I

should stay as far away from the experimental side of systems biology as possible!

I was fortunate enough to be a member of Christ Church (college) throughout my time as a graduate student. Hanging out in the GCR frequently provided a welcome distraction from the technical details associated with such research. In particular, thanks go to Drew Foxall and Euan Wielewski for proving that Oxford can be a normal place (at least some of the time) and to Kerrith Davies for striving to prove the opposite.

My journey down this road began at the University of Reading in the School of Systems Engineering. Without the guidance of Victor Becerra I certainly would never have come to Oxford. Victor supervised me as a masters student and sparked my interest in control theory and optimization. He also gave me the chance to get involved in research by hiring me as an RA for a year, an experience I will always be grateful for.

Finally I dedicate this thesis to my family- Mum, Dad, Alex and Helen. Thanks for everything but especially for putting up with the missed phone calls, emails and general excuses of being too busy while writing this thing up! Thank you.

Abstract

This thesis is concerned with investigating new methods for the analysis of large-scale dynamical systems using convex optimization. The proposed methodology is based on composite Lyapunov theory and is computationally implemented using polynomial programming techniques. The main result of this work is the development of a system decomposition framework that makes it possible to analyze systems that are of such a scale that traditional methods cannot cope with.

We begin by addressing the problem of model invalidation. A barrier certificate method for invalidating models in the presence of uncertain data is presented for both continuous and discrete time models. It is shown how a reparameterization of the time dependent variables can improve the numerical conditioning of the underlying optimization problem.

The main contribution of this thesis is the development of an automated dynamical system decomposition framework that permits us to verify the stability of systems that typically have a state dimension large enough to render traditional computational methods intractable. The underlying idea is to decompose a system into a set of lower order subsystems connected in feedback in such a manner that composite methods for stability verification may be employed. What is unique about the algorithm presented is that it takes into account both dynamics *and* the topology of the interconnection graph.

In the first instance we illustrate the methodology with an ecological network and primal Internet congestion control scheme. The versatility of the decomposition framework is also highlighted when it is shown that when applied to a model of the EGF-MAPK signaling pathway it is capable of identifying biologically relevant subsystems in addition to stability verification.

Finally we introduce stability metrics for interconnected dynamical systems based on the theory of dissipativity. We conclude by outlining a clustering based decomposition algorithm that explicitly takes into account the input and output dynamics when determining the system decomposition.

Contents

Acknowledgements iv

Abstract vi

1 Introduction 1

1.1 Thesis Outline and Contributions 4

2 Background 7

2.1 Notation 8

2.2 Lyapunov Theory 9

2.3 Convex Optimization 12

2.3.1 Semidefinite Programming 14

2.4 Nonnegative Polynomials 16

2.5 The Positivstellensatz 20

2.5.1 Lyapunov Function Construction 25

2.5.2 SOS Scalability Issues 31

2.6 Conclusion 32

3 Model Invalidation Using SOS Programming 33

3.1 Problem Formulation 34

3.2 Parameter Estimation 36

3.2.1 Biochemical Reaction Network Example 41

3.3 Continuous Time Model Invalidation 45

3.3.1 Biochemical Reaction Network Example Revisited 50
3.4 Discrete Time Model Invalidation 53
3.4.1 Population Growth Example 55
3.5 Conclusion 58
3.6 Appendix 59

4 System Decomposition via Composite Lyapunov Functions 60
4.1 Problem Formulation 61
4.1.1 Motivation 62
4.2 Preliminaries 64
4.2.1 Algebraic Graph Theory 64
4.2.1.1 Graph Partitioning 65
4.2.2 Composite Lyapunov Functions 66
4.3 Sparse Lyapunov Functions 70
4.4 Decomposition 72
4.4.1 Graphical Representation of a Dynamical System 79
4.4.1.1 Topology 80
4.4.1.2 Edge weights 80
4.4.2 Decomposition Algorithm 82
4.4.3 Overlapping Decomposition 89
4.5 Remarks on Tractability 92
4.6 Conclusion 94
4.7 Appendix 96

5 Decomposition of the EGF-MAPK Signaling Pathway 98
5.1 Pathway Model 98
5.2 Results 99
5.3 System Decomposition With Uncertainty 102
5.4 Conclusion 107

6 System Decomposition via Dissipation Inequalities **108**

6.1 Problem Formulation . . . 109

6.2 Preliminaries . . . 110

6.2.1 Dissipation Inequalities . . . 110

6.2.2 Small Gain Theorem . . . 113

6.3 $\mathcal{L}_2 \to \mathcal{L}_2$ Performance via Decomposition . . . 115

6.3.1 Input-output Numerical Example . . . 120

6.4 Interconnected Systems . . . 123

6.4.1 Passivity . . . 124

6.4.2 Finite Gain . . . 126

6.4.3 Input Strong Passivity . . . 129

6.4.4 RC Network Example . . . 131

6.5 A Clustering Based Decomposition Algorithm . . . 132

6.5.1 Appending states to LTI systems . . . 133

6.5.1.1 Lower bound estimates . . . 134

6.5.1.2 Upper bound estimates . . . 138

6.5.1.3 Clustering Algorithm . . . 142

6.5.1.4 Input-Output Clustering . . . 145

6.6 Further Remarks on Tractability . . . 152

6.6.1 Optimization over Π . . . 152

6.7 Conclusion . . . 158

6.8 Appendix . . . 159

7 Conclusions **161**

7.1 Summary . . . 161

7.2 Future research directions . . . 163

Notation **165**

References **166**

List of Figures

2.1 The Motzkin polynomial in 2 variables 18

2.2 Vector field of system (2.27) and associated Lyapunov function level curves 28

2.3 RoA estimate for (2.27) 30

2.4 Improved RoA Estimate 31

3.1 Parameter estimation example 44

3.2 Invalid model example 46

3.3 Barrier certificate example 52

3.4 Discrete-time population growth model 55

3.5 Data and model at $k = 2$ and $k = 3$. 56

4.1 Network topology for Internet example 86

4.2 Decomposed Lotka-Volterra system 89

5.1 EGF-MAPK pathway 100

5.2 Decomposed EGF-MAPK pathway 102

6.1 Small-gain feedback interconnection 115

6.2 Decomposed non-autonomous dynamical system block diagram . . 118

6.3 RC network example 131

6.4 Allowable state configurations for appended systems 134

6.5 Node Clustering Example 144

6.6 Possible clusters at the first iteration of the clustering algorithm. . 145

6.7 LTI system cascade 146
6.8 Feedback block diagram 149

List of Tables

4.1 Computational cost of constructing a Lyapunov function. 89

4.2 The Overlapping Decomposition Algorithm 90

Chapter 1

Introduction

One of the most fundamental problems in control theory is that of verifying the stability properties of an equilibrium point of a dynamical system. This seemingly innocent question has challenged researchers for years. As computational power increases at an astonishing rate and sensors become more accurate, researchers are continually raising the stakes by developing increasingly intricate and complex system models.

Mathematical biology [1] and the related field of systems biology [2] in particular, have rapidly accelerated over the past decade, thanks in part to advances in high throughput experimental techniques. There is now a wealth of models of biological systems. In many cases multiple models of the same phenomena exist. Typically these models consist of large sets of coupled nonlinear differential equations whose behaviour is characterized in part by large parameter vectors. Inspired by models as complex as those in the biological literature the work in this thesis aims to provide theoretically sound and computationally scalable algorithms for model invalidation, stability and performance analysis.

At present algorithmic methods for the analysis of nonlinear systems is possible for systems comprising of only a modest number of states. We acknowledge that there is no concrete definition for the label "*large-scale*" when used to describe a dynamical system. Indeed the term *large-scale* is heavily dependent on

context and different researchers will have different interpretations of what it means. For the purpose of this thesis we will assume that the term large-scale is dependent on two properties: i) state dimension and ii) system description, e.g. linear, nonlinear, vector field degree, uncertain, highly coupled etc. A system of state dimension n and of a given type is called large-scale if it is not computationally tractable to verify a desired system property using convex optimization.

For example, nonlinear dynamical systems described by polynomial differential equations can be analyzed using sum of squares programming which can be formulated as a convex optimization problem. Assuming non-trivial coupling in the dynamics, systems with up to approximately 7-states can be analyzed before the optimization programme becomes too large to solve and suffers from numerical ill-conditioning. Systems of this type with a larger state dimension are thus termed large-scale dynamical systems. We acknowledge that this is not a traditional notion of large-scale and that many control theorists would not consider a 7-state system to be large-scale. However this definition provides a natural measure of system size suitable for the work described in this thesis.

Computational stability analysis is made even more difficult when one considers uncertainty in the parameters that describe the system. Unfortunately, any realistic system model must take uncertainty into account. With regards to biological models, uncertainty can arise in a number of ways. For example most kinetic constants in a biochemical reaction network cannot be measured, furthermore, for cases when measurements can be made they are clearly subject to experimental error.

What is desirable, and ultimately what the work in this thesis aims to provide is a toolbox of scalable algorithms for the stability analysis of large-scale dynamical systems. It is not only important to be able to verify the stability properties of a given system model but also to consider the stability of networks of these systems [3]. The primary focus of this thesis is to develop stability algorithms for

systems of the form

$$\dot{x}(t) = f(x(t), p), \quad x(0) = x_0 \in \mathcal{X}$$

where $f(0, p) = 0$, $p \in \mathcal{R}^m$ is a parameter vector, $x(t) \in \mathcal{R}^n$ is the state vector and n is "*large*".[1] Stability verification of the equilibrium point $x^* = 0$ refers to determining if all trajectories of the system system starting close to the equilibrium point tend to the origin. One solution to this problem is to solve the system of differential equations and obtain an analytic expression for $x(t)$. In general this approach is not possible for highly coupled nonlinear systems, fortunately as demonstrated by the Russian mathematician A.M. Lyapunov, neither is it necessary, see for example [4]. Instead, if a function (known as a Lyapunov function) can be shown to exist that satisfies certain positivity properties on $\mathcal{X}$ then stability of the equilibrium point x^* is guaranteed.

Consider the case where the vector field f is linear and the parameters p are fixed and known. This corresponds to the linear system $\dot{x}(t) = Ax(t)$, whose stability is guaranteed if and only if there exists a positive definite matrix P of appropriate dimensions such that $A^T P + PA \prec 0$. This is an example of a Linear Matrix Inequality (LMI) which can be solved efficiently (in polynomial time with respect to state dimension) using convex optimization [5].

When f is not a linear map but instead a nonlinear vector field general algorithms for constructing Lyapunov functions do not exist. However, if we restrict our attention to the case where f is a vector of polynomial functions, then thanks to advances in the seemingly unrelated fields of convex optimization [6] and real algebraic geometry [7] it is possible to search for a polynomial Lyapunov function by solving a convex optimization problem [8] called a semidefinite programme [9]. The methodology is based on computing the sum of squares decomposition of a given polynomial. One of the open challenges in this direction, and a central

[1] It is assumed that f is sufficiently smooth so as to guarantee local existence and uniqueness of solutions.

theme of this thesis is addressing the poor scalability of sum of squares programming techniques.

1.1 Thesis Outline and Contributions

The main results of this thesis are concerned with developing scalable algorithms for the analysis of large-scale dynamical systems with applications in engineering and mathematical biology. In particular, algorithms for decomposing dynamical systems into subsystems of lower dimension with favourable properties are developed. Such problems are generally combinatorial in nature and as a result are typically $\mathcal{NP}$-hard. For this reason it is desirable to formulate approximate solutions which are simpler to compute.

We begin by introducing convex optimization formulations of various system properties such as passivity and stability. For the case of nonlinear systems, these formulations are only tractable for systems with a small state dimension. The remainder of the thesis uses the computational machinery introduced to address: i) The problem of model (in)validation in systems biology. Here we emphasize the increase in computational complexity as the system complexity increases. ii) The design of scalable optimization based stability algorithms for analyzing large-scale dynamical systems. The main contribution is the introduction of an automated system *decomposition* framework for stability and performance analysis. Below the contents of each chapter are summarized and peer reviewed papers that have appeared in the literature are listed. The necessary background material and relevant literature are presented at the beginning of each chapter.

- In Chapter 2, the notion of Lyapunov stability for dynamical systems is reviewed. We treat both the linear and nonlinear case and describe how both system types can be analyzed computationally by formulating analysis questions as convex optimization problems. The properties of nonnegative polynomials and the sum of squares decomposition are presented as well as

a result from real algebraic geometry known as the Positivstellensatz. Key ideas are illustrated by numerical examples.

- In Chapter 3, the computational tools introduced in the previous chapter are illustrated on problems arising in mathematical biology. We consider the problem of model invalidation and parameter estimation. It is shown how sum of squares programming can be used to obtain a lower bound on the quality of the fitted model to the observed data. Building on these results we then show how a continuous-time nonlinear model containing uncertainty can be invalidated using *barrier certificates* and a discrete-time system can be invalidated using the Positivstellensatz. The invalidation process is often overlooked in the system analysis and design cycle as it is typically difficult to automate and simulation based approaches are often inconclusive. In this chapter the methods developed do not require *any* simulations and can provide definitive answers. Our solution to the discrete-time invalidation is novel, whereas the main result with regards to the continuous-time case is a re-parametrization of the time dependency which allows for improved computational performance and paves the way for invalidation of larger, more complex models. The results of this chapter were published in [10].

- In Chapter 4, we address the problem of how to analyze the stability of large-scale autonomous dynamical systems. A *system decomposition* framework is presented that aims to break a large system into subsystems of smaller dimension connected in feedback. The driving idea is that stability of the original system can be verified through a subsystem-by-subsystem analysis which incurs a lower computational cost. The subsystem stability certificates (Lyapunov functions) can then be combined to provide a composite Lyapunov function for the original system. It is shown that in the linear case, decomposing a system by minimizing the energy flow between states facilitates the composite approach. Additionally, a numerical method based

on maximizing sparsity is presented that helps reduce the computational complexity associated with solving large sum of squares optimization problems. The results presented in this chapter have appeared in [11–13].

- In Chapter 5 the versatility of the decomposition algorithm is illustrated on a model of the Epidermal Growth Factor (EGF) receptor-induced MAP kinase signalling pathway. The model is a large-scale nonlinear dynamical system that contains parametric uncertainty. We show that the decomposition algorithm is capable of i) identifying relevant subsystems ii) verifying stability of the subsystems and iii) verifying stability of the composite model. A method for handling uncertainty in the model is also presented for the case when the dynamical system can be written in Linear Parameter Varying (LPV) form. This example appeared in [14].

- In chapter 6 dissipation inequalities and system decomposition methods are presented in order to analyze input-output stability of interconnected dynamical systems. We begin by presenting a method for obtaining a bound on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of a large-scale nonlinear dynamical system. We then focus on Linear Time Invariant system stability using quadratic storage functions and how the stability of networks of such systems can be verified. Finally we look at how the gain of a system changes as states are appended up-stream, down-stream and in feedback. These results are then incorporated into a *bottom-up* clustering-decomposition algorithm. Results from this chapter have appeared in [15].

- Finally, in Chapter 7 the thesis is concluded and future research directions are presented.

Chapter 2

Background

In this chapter the mathematical and computational tools that will be used throughout this thesis are introduced. The techniques presented are assembled from a variety of fields including control theory, dynamical systems, optimization and real algebraic geometry. In particular, Semidefinite Programmes (SDPs)[1], which are a special class of *convex optimization* problem will be used to formulate and solve all systems analysis questions in this work. When we are concerned with analyzing linear dynamical systems the algorithms we develop will take the form of an SDP. However, when the system of interest is governed by nonlinear differential equations the algorithms will take the form of sum of squares programmes. Such programmes are formulated and solved as SDPs, this parsing process can be done manually for small problems as we will illustrate, however for the general case the MATLAB toolbox SOSTOOLS [16] automates this process.

We begin by introducing in Section 2.2 the main theoretical tool for systems analysis in this thesis, namely the Lyapunov framework. Computational methods for constructing Lyapunov functions for linear systems using convex optimization are presented in Section 2.3. For nonlinear systems no algorithmic method for Lyapunov function construction exists in the general case. However, if one consid-

[1]The abbreviation SDP will be used to refer to both a semidefinite programme and semidefinite programming.

ers only polynomial systems then the sum of squares decomposition [8] presented in Section 2.4 can be used to algorithmically generate a Lyapunov function as described in Section 2.5.1.

2.1 Notation

Throughout this thesis we adopt the following notation:

Vector Spaces: Let $\mathcal{R}^n$ denote the n-dimensional Euclidean space, then $\mathcal{R}^{n\times m}$ denotes the space of $n \times m$ matrices with real elements. The space of symmetric $n \times n$ matrices is denoted by $\mathcal{S}^n$. If $M \in \mathcal{S}^n$ then $M \succ 0$, $M \succeq 0$, denotes that M is positive definite and positive semidefinite respectively. The set of all $n \times n$ positive definite and positive semidefinite matrices are denoted by $\mathcal{S}^n_{++}$ and $\mathcal{S}^n_{+}$. The column vector $[1, \ldots, 1]^T$ is denoted by $\mathbf{1}$, the dimension will be made clear from the context.

Signal and System Norms: Given a matrix $A \in \mathcal{S}^n$, the maximum and minimum eigenvalues of A are denoted by $\overline{\lambda}(A)$ and $\underline{\lambda}(A)$ respectively. For $X \in \mathcal{R}^{n\times m}$, $\overline{\sigma}(X)$ and $\underline{\sigma}(X)$ denote the maximum and minimum singular values of X. For a vector $x \in \mathcal{R}^n$ we use the standard p-norms on $\mathcal{R}^n$ denoted by $\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$ with $1 \leq p \leq \infty$. For the case of $p = 2$ we drop the subscript and simply use $\|x\|$. For a piecewise continuous, square integrable function u of dimension n on the domain $[0, \infty)$ we will make use of the $\mathcal{L}^n_2$ norm defined by

$$\|u\|_{\mathcal{L}^n_2} = \sqrt{\int_0^\infty u^T(t)u(t)dt} \leq \infty.$$

When it is clear from context the superscript denoting the dimension will be dropped. The matrix $M \in \mathcal{R}^{n\times m}$ defines a mapping $y = Mx$ from $\mathcal{R}^m$ to $\mathcal{R}^n$.

The *induced* p-norm denoted $\|M\|_p$ is defined by

$$\|M\|_p = \sup_{x \neq 0} \frac{\|Mx\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|Mx\|_p.$$

For the special case of $p = 2$ we drop the subscript notation. The induced 2-norm is called the spectral norm[1] and is given by $\|M\| = \sqrt{\overline{\lambda}(M^T M)} = \overline{\sigma}(M)$.

Real Algebraic Geometry: Denote by $\mathcal{R}_{n,r}[x]$ the ring of commutative polynomials of maximum degree r in the variable x where $x \in \mathcal{R}^n$. When the dimension of x is clear we will simply refer to $\mathcal{R}_r[x]$. Furthermore, when the degree bound is not specified, or it is clear from context then we use the notation $\mathcal{R}[x]$. Sum of squares polynomials of maximum degree r in the variable x are denoted by $\Sigma_{n,r}[x]$ in the same manner as above. For a polynomial $p(x) \in \mathcal{R}[x]$ we denote the degree of p by $\partial(p)$.

The precise definitions will be introduced where needed. A full list of the notation used is given in Appendix A.

2.2 Lyapunov Theory

In this section we address the issue of stability analysis of a system described by a set of n coupled first-order Ordinary Differential Equations (ODEs). A typical system of ODEs takes the form:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1, \ldots, x_n, t) \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1, \ldots, x_n, t) \end{aligned} \tag{2.1}$$

[1] When M is normal ($MM^T = M^T M$) the spectral radius of M is equal to the induced 2-norm.

where $x_i(t) \in \mathcal{R}$ are the state variables and t represents time. More compactly the ODE system (2.1) can be written using vector notation as

$$\dot{x}(t) = f(x, t) \tag{2.2}$$

where $x(t) \in \mathcal{R}^n$, $f : \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}^n$ and the initial condition is $x(0) = x_0$. We will focus on the case where the vector field f is not a function of time; such systems are termed autonomous systems and are written as

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

Let $\mathcal{X} \subset \mathcal{R}^n$ be the domain of interest, we assume that $f : \mathcal{X} \to \mathcal{R}^n$ is locally Lipschitz in $\mathcal{X}$. Without loss of generality we assume that $f(0) = 0$, i.e. the origin, is the equilibrium point of interest, this is realised through a simple change of coordinates.

For the case of autonomous systems of the form (2.3) we are interested in determining the stability of the equilibrium point $x = 0$. We are now ready to state two formal definitions of stability:

Definition 1 (Stability [4]) *The equilibrium point $x = 0$ of (2.3) is*

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

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \quad \forall\ t \geq 0$$

- *asymptotically stable if it is stable and δ can be chosen such that*

$$\|x(0)\| < \delta \Rightarrow \lim_{t \to \infty} x(t) = 0$$

- *unstable if it is not stable.*

The stability criteria above state that all trajectories starting in a δ-neigbourhood of the origin cannot leave the ϵ-neighbourhood. Fortunately, these properties can be verified without explicitly solving the system of equations (2.3) if it can be

shown that an *energy-like* function exists and satisfies certain properties. Such functions are called Lyapunov functions. Before presenting Lyapunov's theorem the following definition is required: Given a differentiable function $H : \mathcal{R}^m \to \mathcal{R}$ define

$$\frac{\partial H}{\partial x}(x) \triangleq \left[\frac{\partial H}{\partial x_1}(x) \ \ldots \ \frac{\partial H}{\partial x_m}(x)\right].$$

Theorem 1 ([4]) *Let $x = 0$ be an equilibrium point of (2.3) and $\mathcal{X} \subset \mathcal{R}^n$ be a region around the origin in which f is Lipschitz. If there exists a continuously differentiable function $V : \mathcal{X} \to \mathcal{R}$ such that the following are satisfied:*

1. *$V(x) > 0$ for all $x \in \mathcal{X} \setminus \{0\}$ and $V(0) = 0$*
2. *$-\dot{V}(x) = -\frac{\partial V}{\partial x} f(x) \geq 0$ for all $x \in \mathcal{X}$*

then the origin is a stable equilibrium. Further more if $-\dot{V}(x) > 0$ for all $x \in \mathcal{X} \setminus \{0\}$ then the origin is asymptotically stable.

A continuously differentiable function $V(x)$ that satisfies the inequality constraints in Theorem 1 is called a Lyapunov function. Consider now a special form of system (2.3) – a Linear Time Invariant (LTI) system:

$$\dot{x} = Ax, \quad x \in \mathcal{R}^n. \tag{2.4}$$

Consider the quadratic Lyapunov function $V(x) = x^T P x$ with $P \succ 0$. The time derivative of $V(x)$ along the trajectories of (2.4) is

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} \\ &= x^T (A^T P + PA) x \\ &= -x^T Q x \end{aligned}$$

where $Q \in \mathcal{S}^n$ and is defined by

$$-Q = A^T P + PA. \tag{2.5}$$

Equation (2.5) is called the *Lyapunov Equation*. If there exists a positive definite matrix Q that satisfies (2.5) then stability of the LTI system (2.4) is verified through Theorem 1. Alternatively, for any matrix $Q \succ 0$, solve (2.5) for P, if and only if (2.4) is stable then we have that $P \succ 0$. Furthermore P will be unique.

In the following section a Linear Matrix Inequality (LMI) [5] formulation for determining stability in the sense of Lyapunov will be presented. LMIs are the constraints in semidefinite optimization programmes which have many desirable properties.

2.3 Convex Optimization

While convex optimization is a subject in its own right, a few important definitions and properties will now be described. For a detailed discussion of the subject the reader is directed to [6] and the references therein. An in depth look at convex optimization techniques for robust control is given in [17].

We begin by introducing the idea of convex sets. Intuitively a vector space $\mathcal{C} \subseteq \mathcal{R}^n$ is convex if for all vectors x and y in $\mathcal{C}$, all points on the connecting line segment between x and y are also in $\mathcal{C}$. A more formal definition follows: Given two vectors $x, y \in \mathcal{R}^n$, parametrically the line through x and y is given by

$$f(\theta) = \theta x + (1 - \theta)y$$

with $\theta \in \mathcal{R}$. Now define the line segment between x and y by

$$[x, y] = \{f(\theta) \mid \theta \in [0, 1]\},$$

then the proceeding definition of a convex set follows naturally:

Definition 2 (Convex Sets) *The subset $\mathcal{C} \subseteq \mathcal{R}^n$ is convex if $[x, y] \subset \mathcal{C}$ for all $x, y \in \mathcal{C}$.*

Some important convex sets that are frequently used in this thesis include Euclidean balls, ellipsoids and the positive semidefinite cone. The next object of interest is the convex function.

Definition 3 (Convex Functions [6]) *A function $f : \mathcal{C} \to \mathcal{R}$ is convex if $\mathcal{C}$ is a convex set and*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \tag{2.6}$$

for all $x, y \in \mathcal{C}$ and all $\theta \in [0, 1]$.

A generic optimization problem can be written as

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) < 0, \quad \text{for } i = 1, \ldots, m \end{aligned} \tag{2.7}$$

In this instance the vector $x \in \mathcal{R}^n$ is the decision vector. The function $f_0(x) : \mathcal{R}^n \to \mathcal{R}$ is the objective function and the functions $f_i(x) : \mathcal{R}^n \to \mathcal{R}$ for $i = 1, \ldots, m$ impose inequality constraints on the decision vector. The optimal decision vector is denoted by x^*.

The optimization problem defined in (2.7) is very general. If the functions $f_0, \ldots, f_m$ are affine then the problem is called a *linear programme*. A function is linear if

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y) \tag{2.8}$$

for all $x, y \in \mathcal{R}^n$ and $\alpha, \beta \in \mathcal{R}$. We are interested in a more general class of *convex optimization problems*. A convex optimization has the form of (2.7) where the functions $f_0, \ldots, f_m$ are convex. Examining (2.8) and (2.6) it is clear that the linear case is more restrictive than the convex case as the *inequality* in (2.6)

is only required to hold for all $\theta \in [0, 1]$ in comparison to the *equality* condition in (2.8) which is enforced for all $\alpha, \beta \in \mathcal{R}$.

Formulating systems analysis questions as convex optimization problems is desirable for many reasons. In particular convex problems do not contain any local minima thus a globally optimal solution is always attainable. Secondly the computational complexity of solving such problems using interior-point methods [6, 18] has been shown to not exceed a polynomial of the problem dimensions [19].

2.3.1 Semidefinite Programming

A Semidefinite Programme (SDP) [9] is a convex optimization problem with a linear objective function and Linear Matrix Inequality (LMI) constraints. The general form of a *primal* SDP with a decision vector $x \in \mathcal{R}^n$ is

$$\begin{aligned} \text{minimize} \quad & c^T x && (2.9) \\ \text{subject to} \quad & F(x) \succeq 0, && (2.10) \end{aligned}$$

where $F(x)$ is an LMI in the given matrices $F_0, \ldots, F_m \in \mathcal{S}^n$ defined by

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i. \tag{2.11}$$

Note that SDPs provide a very flexible optimization framework and easily allow for the incorporation of affine constraint sets and multiple LMI constraints. The following SDP has both LMI and affine constraints. Given $c \in \mathcal{R}^n$, $F_0, \ldots, F_m \in \mathcal{S}^n$, $A \in \mathcal{R}^{m \times n}$ and $b \in \mathcal{R}^m$ solve

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & F(x) \succeq 0 \\ & Ax - b \geq 0 \end{aligned} \tag{2.12}$$

where the affine inequality is taken element-wise. The constraints in SDP (2.12) are equivalently written as an LMI

$$\begin{bmatrix} F(x) & 0 \\ 0 & H(x) \end{bmatrix} \succeq 0$$

where $H(x)$ is a diagonal $m \times m$ matrix with $H_{ii} = a_i^T x - b_i$ for $i = 1, \ldots, m$ where a_i denotes the i^{th} row of A. There are many freely available SDP solvers available such as SeDuMi [20], CVX [21] and SDP3 [22]. Unless otherwise stated we will use the SeDuMi solver in conjunction with the LMI parser YALMIP [23] to solve all SDP optimization problems.

One of the most common LMIs that arises in control and dynamical systems theory is that used to construct quadratic Lyapunov functions for stability verification of linear systems of the form (2.3). This corresponds to searching for a matrix $P = P^T$ such that

$$P \succ 0, \quad A^T P + PA \prec 0.$$

The Lyapunov function is then given by $V(x) = x^T P x$. A more involved example of an SDP that arises in control theory (and which will be used in Chapter 6) is determining the *dissipation rate* of a dynamical system. The system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

is said to have a dissipation rate of ϵ if

$$\int_0^T (u^T y - \epsilon u^T u) dt \geq 0, \quad x(0) = 0 \tag{2.13}$$

for all $T \geq 0$. The maximal rate of dissipation is given by the largest ϵ that

satisfies (2.13). Taking a Lyapunov based approach, assume that there exists a quadratic function $V(x) = x^T P x$ with $P \succ 0$ such that $\dot{V}(x) \leq 2u^T y - 2\epsilon u^T u$. Integrating this expression from 0 to T and noting that by construction $V(x(T)) \geq 0$ we conclude that (2.13) must hold. It is easy to show that the derivative inequality can be represented by the LMI

$$\begin{bmatrix} A^T P + PA & PB - C^T \\ B^T P - C & 2\epsilon I - D^T - D \end{bmatrix} \preceq 0. \tag{2.14}$$

To determine the maximum dissipation, ϵ is maximized subject to (2.14).

2.4 Nonnegative Polynomials

In the previous section Theorem 1 illustrated that verifying the stability of a system can be reduced to checking that two inequalities satisfy non-negativity conditions. Unfortunately, the central object of interest in this work – *nonlinear dynamical systems* make testing these non-negativity constraints very difficult. For reasons that will become clear, we restrict our attention to systems described by polynomial vector fields. In this case it has been shown that testing polynomial non-negativity when the degree of the polynomial is greater than or equal to 4 is an $\mathcal{NP}$-hard problem [24]. Motivated by this difficulty, tractable computational relaxations to polynomial non-negativity based on the sum of squares (SOS) decomposition have received a lot of attention. Here we introduce the main ideas behind SOS programming that will be used throughout this work.

One of the central problems of real algebraic geometry [7] is to determine if a polynomial with real coefficients attains non-negative values on a closed semialgebraic set. Such polynomials are termed *positive*. We begin with a few definitions.

Let $\mathcal{R}[x] \triangleq \mathcal{R}[x_1, \ldots, x_n]$ be the commutative ring of polynomials in $(x_1, \ldots, x_n)$ with real coefficients.

Definition 4 (Monomial) *Given an n-tuple of non-negative integers $(\alpha_1, \ldots, \alpha_n)$, a monomial in $x_1, \ldots, x_n$ takes the form $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$. The degree of a monomial is given by $|\alpha| = \sum_{i=1}^n \alpha_i$.*

Definition 5 (Polynomial) *A polynomial f in $x_1, \ldots, x_n$ with coefficients in $\mathcal{R}$ is a finite linear combination of monomials written in the form*

$$f(x) = \sum_{\alpha} a_\alpha x^\alpha, \quad a_\alpha \in \mathcal{R}.$$

The total degree of f, denoted $\partial(f)$, is the maximum $|\alpha|$ corresponding to a nonzero a_α.

Definition 6 (Form) *Let p be a polynomial consisting of m nonzero monomials $\alpha^1, \ldots, \alpha^m$. If $|\alpha^1| = |\alpha^2| = \cdots = |\alpha^m|$ then p is called a form. The set of nonzero forms of degree r in n variables is denoted $\mathcal{F}_{n,r}$.*

The following definition of an SOS polynomial and the relationship between convex optimization and SOS polynomials will be used heavily throughout this thesis.

Definition 7 (SOS Polynomial) *A polynomial $P \in \mathcal{R}[x]$ is a sum of squares (SOS) if it can be decomposed into N polynomials and expressed as*

$$P(x) = \sum_{i=1}^{N} p_i^2(x) \tag{2.15}$$

where $p_i, \ldots, p_N \in \mathcal{R}[x]$. The set of SOS polynomials of degree m is denoted by $\Sigma_m[x]$.

It is clear from (2.15) that SOS polynomials are nonnegative on $\mathcal{R}^n$ i.e. $P(x) \geq 0$ for all $x \in \mathcal{R}^n$. Necessarily it follows that $\partial(P)$ must be even. Given this definition of SOS polynomials and their nonnegativity properties two questions immediately arise; i) Are all globally nonnegative polynomials SOS polynomials? and ii) How does one compute a sum of squares decomposition?

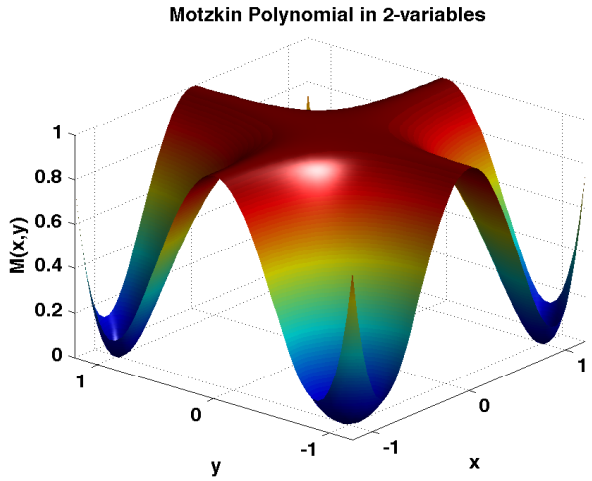


Figure 2.1: The Motzkin polynomial in 2 variables

The answer to the first question is no. There do exist globally nonnegative polynomials that cannot be represented by sums of squares. However, there are three special cases of nonnegative polynomials, identified in 1888 by David Hilbert [25], that can always be represented as a sum of squares. They are: polynomials of one variable, polynomials of degree two in any number of variables, and polynomials of degree four in three variables.

A classic example of a polynomial that is nonnegative but does not admit a sum of squares representation is the *Motzkin* polynomial given below in 2 variables, see Figure 2.1:

$$M(x, y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1.$$

Related work has looked at trying to quantify the difference in volume of the cones of nonnegative polynomials and SOS polynomials [26]. In an attempt to bridge the gap between these two cones Reznick [27] showed that if $f(x) > 0$ on $\mathcal{R}^n$ then

$$f(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$$

is a sum of squares.

The most common method for computing a sum of squares decomposition is the so called *Gram matrix* method. One of the first papers to use this method is [28], however there is no mention of convexity in the solution. Indeed it was the seminal work of Pablo Parrilo [8] that made clear the connection between convexity and geometry. In the following theorem a reference is made to *sparse* polynomials. Informally a polynomial is said to be sparse if many of the monomial coefficients are equal to zero. This notion of sparsity will be made explicit in Chapter 4.

Theorem 2 ([29]) *The existence of a sum of squares decomposition of a polynomial in n variables of degree $2d$ can be decided by solving a semidefinite programming feasibility problem. If the problem is dense (no sparsity), the dimensions of the matrix inequality are $\binom{n+d}{d} \times \binom{n+d}{d}$.*

The following proposition and numerical example illustrate the Gram matrix method and make precise the SDP representation of SOS polynomials:

Proposition 1 *A polynomial $p \in \mathcal{R}[x]$ of degree $2d$ is a sum of squares if and only if there exists a positive semidefinite matrix Q and vector of monomials $Z(x)$ containing monomials in x of degree less than or equal to d such that*

$$p(x) = Z(x)^T Q Z(x). \tag{2.16}$$

The following example taken from [29]:

Example 1 *Given the fourth order polynomial (form) in two variables, $p(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$ determine if it is a sum of squares.*

Step 1: Define the vector of monomials $Z(x)$, remembering that the monomials in $Z(x)$ must be of degree ≤ 2. Here we choose $Z(x) = [x_1^2 \quad x_2^2 \quad x_1x_2]^T$ as $p \in \mathcal{F}_{2,4}$.

Step 2: Expand $Z^T(x)QZ(x)$:

$$Z(x)^T Q Z(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$$

$$= q_{11}x_1^4 + q_{22}x_2^4 + (2q_{12} + q_{33})x_1^2 x_2^2 + 2q_{13}x_1^3 x_2 + 2q_{23}x_1 x_2^3$$

Step 3: Equating the coefficients from the expansion with the coefficients of $p(x_1, x_2)$ results in the following affine relations:

$$q_{11} = 2, \quad q_{22} = 5, \quad q_{13} = 1, \quad q_{23} = 0, \quad 2q_{12} + q_{33} = -1$$

The resulting SDP is a search for coefficients for q_{12}, q_{33} that satisfy the last affine relationship above subject to the constraint that $Q \succeq 0$. For this example $q_{12} = -3$ and $q_{33} = 5$ solve the problem. Performing a Cholesky factorization *[30]* of Q we obtain

$$Q = L^T L, \quad \textit{with } L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

which leads to the SOS decomposition:

$$p(x) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1 x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1 x_2)^2. \qquad \blacksquare$$

Throughout this work the freely available MATLAB toolbox SOSTOOLS [16] is used in conjunction with SeDuMi to compute all SOS decompositions.

2.5 The Positivstellensatz

One of the main advantages of the sum of squares decomposition is that it can be used as a computational relaxation to polynomial nonnegativity. It has al-

ready been shown in Section 2.2 why this is desirable, however, the results in the previous sections are global i.e. they hold for all of $\mathcal{R}^n$. In many instances it is desirable for these results to only hold on a given domain $\mathcal{D} \subseteq \mathcal{R}^n$. A standard example is that of trying to find a Lyapunov function for a nonlinear system that contains multiple equilibrium points. In this case it is clear that the Lyapunov inequalities are only required to hold over a given set. Similar arguments appear many times in robust control theory.

A result from real algebraic geometry [7] called the Positivstellensatz, which we will now introduce, provides a feasibility test for determining the nonnegativity of real polynomials over semialgebraic sets. The Positivstellensatz establishes a relationship between geometric objects (*affine varieties*) and the validity of an algebraic relationship (*polynomial ideal*) [31].

The domain over which we wish to determine the nonnegativity of a given polynomial is defined by a set of polynomial equations called a semialgebraic set:

Definition 8 (Semialgebraic Set) *The basic closed semialgebraic set is a subset of $\mathcal{R}^n$ defined by a finite number of polynomial inequalities*

$$\{x \in \mathcal{R}^n \mid f_i(x) \geq 0,\ f_i \in \mathcal{R}[x], \forall i = 1, \ldots, j\}\,. \tag{2.17}$$

Definition 9 (Ideal) *Given the multivariate polynomials $\{g_1, \ldots, g_m\} \in \mathcal{R}[x]$, the Ideal generated by g_i for $i = 1, \ldots, m$ is the set*

$$\mathcal{I}(g_1, \ldots, g_m) = \left\{ \sum_{i=1}^{m} t_i g_i \mid t_1, \ldots, t_m \in \mathcal{R}[x] \right\}.$$

Definition 10 (Multiplicative Monoid) *Given polynomials $\{h_i, \ldots, h_u\} \in \mathcal{R}[x]$ the Multiplicative Monoid, denoted $\mathcal{M}(h_1, \ldots, h_u)$, generated by h_i for $i = 1, \ldots, u$ is the set of all finite products of h_i including 1.*

Definition 11 (Cone) *Given the multivariate polynomials $\{f_1, \ldots, f_s\} \in \mathcal{R}[x]$*

the Algebraic Cone generated by f_i *for* $i = 1, \ldots, s$ *is the set*

$$\mathcal{C}(f_1, \ldots, f_s) = \left\{ s_0 + \sum_{i=1}^{r} s_i F_i \mid F_i \in \mathcal{M}(f_1, \ldots, f_m),\ s_i \in \Sigma[x] \right\}$$

where r *denotes the number polynomials in* $\mathcal{M}$ *and* $\Sigma[x]$ *denotes the set of sum of squares polynomials in* x*.*

Using the preceding definitions we can now state the Positivstellensatz theorem:

Theorem 3 (Positivstellensatz) *Let* $\{f_1, \ldots, f_s\}$, $\{g_1, \ldots, g_m\}$ *and* $\{h_1, \ldots, h_u\}$ *be finite multivariate polynomials in* $\mathcal{R}[x]$*. Denote by* $\mathcal{C}$ *the algebraic cone generated by* $\{f_1, \ldots, f_s\}$*,* $\mathcal{I}$ *the ideal generated by* $\{g_1, \ldots, g_m\}$ *and* $\mathcal{M}$ *the multiplicative monoid generated by* $\{h_1, \ldots, h_u\}$*. Then the following statements are equivalent:*

- *The set*

$$\left\{ x \in \mathcal{R}^n \;\middle|\; \begin{array}{l} f_1(x) \geq 0, \ldots f_s(x) \geq 0 \\ g_1(x) = 0, \ldots, g_m(x) = 0 \\ h_1(x) \neq 0, \ldots, h_u(x) \neq 0 \end{array} \right\} \tag{2.18}$$

  is empty.

- *There exist* $f \in \mathcal{C}$*,* $g \in \mathcal{I}$ *and* $h \in \mathcal{M}$ *such that*

$$f + g + h^2 = 0. \tag{2.19}$$

For a more detailed treatment and the proof of the Positivstellensatz (and algebraic geometry) see [7].

It can be seen that the Positivstellensatz establishes an equivalence relationship between a geometric object (2.18) and an algebraic relationship (2.19). Notice that there is no mention in Theorem 3 of the degree bounds that should be placed on the sum of squares multipliers, s_i that define $\mathcal{C}$ or polynomials t_i

that define $\mathcal{I}$. Typically one places an upper bound on the degree of each s_i and a hierarchy of set-emptiness tests (2.18) are considered. A systems theoretic interpretation of the Positivstellensatz is that it can be thought of as extending the S-Procedure [5, 32] from quadratic functions and positive scalar multipliers to polynomial functions and SOS multipliers. Of course the analogy is not exact as the S-procedure is sufficient but not necessary for more than two inequalities.

The following examples should make clear the ideas presented above:

Example 2 ([33]) *Consider the system of polynomial relations:*

$$f(x,y) := x - y^2 + 3 \geq 0, \quad h(x,y) := y + x^2 + 2 = 0 \tag{2.20}$$

and prove that there is no $(x,y) \in \mathcal{R}^2$ *such that (2.20) is satisfied. Using Theorem 3, system (2.20) has no solution, if and only if there exist SOS polynomials* $s_1, s_2 \in \Sigma[x,y]$ *and a polynomial* $t \in \mathcal{R}[x,y]$ *that satisfy* $s_1 + s_2 f + th = -1$*. If the left hand side of this expression is evaluated for any feasible solution of (2.20) then the result would be positive, which is a contradiction. The polynomials that certify no solution exists to (2.20) are* $t = -6$*,* $s_2 = 2$ *and* $s_1 = 1/3 + 2(y+3/2)^2 + 6(x-1/6)^2$*.*

In later chapters it will become necessary to solve the *graph partitioning problem*. A more detailed description of the problem and its relevance to this work will be presented when relevant. For now we will simply state the optimization problem and show how the Positivstelensatz can be used to formulate a convex relaxation to this $\mathcal{NP}$-hard problem. Given a positive semidefinite matrix $L \in \mathcal{S}^n_+$ known as the *Laplacian* matrix [34, 35], the graph partitioning problem seeks to find a vector $x \in \mathcal{R}^n$ such that $x_i = \pm 1$ that minimizes the quadratic function $f(x) = \frac{1}{2} x^T L x$. Formally this is stated as:

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T L x \\ \text{s.t.} \quad & x_i^2 = 1, \quad i = 1, \ldots, n \\ & x^T \mathbf{1} \in [-n+1, n-1] \end{aligned} \tag{2.21}$$

where the final constraint ensures that $x \neq \mathbf{1}$ and $x \neq 0$. We can use the Positivstellensatz to obtain a lower bound for (2.21), i.e. find scalar γ such that $f(x) > \gamma$. In particular we would like to find the largest lower bound (maximal γ), i.e. $\inf_{x_i=\pm 1} f(x)$.

The set emptiness test corresponding to the lower bound computation is

$$\left\{ x \in \mathcal{R}^n \;\middle|\; \begin{array}{ll} \frac{1}{2}x^T L x \leq \gamma, & x_i^2 - 1 = 0,\ i = 1, \ldots, n, \\ \frac{1}{2}x^T L x \neq \gamma, & (x^T\mathbf{1} - n + 1)(x^T\mathbf{1} + n - 1) \leq 0 \end{array} \right\} = \emptyset.$$

Let $\delta = 1/2 x^T L x - \gamma$, $\beta = (x^T\mathbf{1} - n + 1)(x^T\mathbf{1} + n - 1)$ and writing the inequalities in the set above in the standard form for use with the Positivstellensatz we obtain the set

$$\left\{ x \in \mathcal{R}^n \;\middle|\; -\delta \geq 0,\ -\beta \geq 0,\ \delta \neq 0,\ x_i^2 - 1 = 0,\ i = 1, \ldots, n \right\} \tag{2.22}$$

which we would like to show is empty.

Applying the Positivstellensatz we obtain the algebraic condition which is equivalent to (2.22):

$$s_0 - s_1\delta - s_2\beta + s_3\delta\beta + \sum_{i=1}^{n} t_i'(x_i^2 - 1) + \delta^{2\kappa} = 0 \tag{2.23}$$

where s_0, s_1, s_2, s_3 are sum of squares polynomials of fixed degree in x and $t_i' = \delta t_i$ where t_i for $i = 1, \ldots, n$ are polynomials of fixed degree in x. Recall that the objective is to find a maximal γ subject to the set emptiness conditions, thus γ is a decision variable as well and so (2.23) is not convex (unless s_1 is fixed). Let $s_0 = 0$, $s_2 = 0$ and set $\kappa = 1$; we now have

$$-s_1\delta + s_3\delta\beta + \sum_{i=1}^{n} \delta t_i(x_i^2 - 1) + \delta^2 = 0,$$

dividing through by δ gives

$$-s_1 + s_3\beta + \sum_{i=1}^{n} t_i(x_i^2 - 1) + \delta = 0,$$

which we rearrange to give:

$$s_3\beta + \sum_{i=1}^{n} t_i(x_i^2 - 1) + \delta \in \Sigma[x].$$

Thus the Sum of Squares optimization problem that obtains the largest lower bound to the graph partitioning problem (2.21) is:

$$\begin{aligned} \max_{\gamma\in\mathcal{R}_+,\, t_i\in\mathcal{R}_{m_i}[x],\, s\in\Sigma[x]} \quad & \gamma \\ \text{s.t.} \quad & \frac{1}{2}x^T Lx - \gamma + s(x)(x^T\mathbf{1} - n + 1)(x^T\mathbf{1} + n - 1) \\ & + \sum_{i=1}^{n} t_i(x)(x_i^2 - 1) \in \Sigma[x]. \end{aligned} \tag{2.24}$$

By fixing the degree of the polynomials $t_i(x)$ and SOS polynomial $s(x)$ a nested set of feasibility tests can be performed, ceasing when the optimization problem returns a feasible solution.

2.5.1 Lyapunov Function Construction

It was shown in Section 2.2 how a Lyapunov function can verify the stability of an equilibrium point of the nonlinear system (2.3). For the case where the system dynamics are linear, stability was shown using an algorithmic methodology involving quadratic Lyapunov functions. We now turn our attention to the case where the vector field f in (2.3) is given by a system of polynomial equations. We further assume that $x = 0$ is the equilibrium of interest and that we wish to verify asymptotic stability.

Step 1: *In accordance with Theorem 1 the region of interest $\mathcal{X}$ must be defined. In order to apply the sum of squares methodology, $\mathcal{X}$ must be representable*

by a semialgebraic set. A simple example is a disc centered at the origin with radius ϕ; $\mathcal{X} = \left\{x \in \mathcal{R}^n, \phi \in \mathcal{R}_{++} \mid x^T x - \phi \leq 0\right\}$ which has an obvious representation by the polynomial inequality $\beta(x)$:

$$\beta(x) \triangleq \sum_{i=1}^{n} x_i^2 - \phi \leq 0.$$

Step 2: *Construct positive definite functions $\varphi_i(x) > 0$, $i = 1, 2$ and define the sum of squares polynomial variables of fixed degree m, $r_i \in \Sigma_{m_i}[x]$ for $i = 1, 2$ and the polynomial variable $V \in \mathcal{R}_p[x]$. Typical choices for $\varphi_i(x)$ are:*

$$\begin{aligned} \varphi_i(x) &= \epsilon\|x\|^2, \quad \epsilon > 0 \\ \varphi_i(x) &= \sum_{i=1}^{n}\sum_{j=1}^{d} \epsilon_{ij} x_i^{2j}, \quad \sum_{j=1}^{m} \epsilon_{ij} > \gamma \quad \forall i = 1, \ldots, n, \quad \gamma > 0, \quad \epsilon_{ij} \geq 0 \quad \forall i, j. \end{aligned}$$

Step 3: *If the following SOS conditions are satisfied:*

$$V(x) + r_1(x)\beta(x) - \varphi_1(x) \in \Sigma[x] \tag{2.25a}$$

$$-\frac{\partial V}{\partial x} f(x) + r_2\beta(x) - \varphi_2(x) \in \Sigma[x] \tag{2.25b}$$

then (2.3) is asymptotically stable on $\mathcal{X}$.

If the SOS inequalities in Step 3 are not satisfied then a hierarchical procedure that increases the degree bounds on the multipliers r_1, r_2 and V can be used. Alternatively, the volume of $\mathcal{X}$ may be decreased by adjusting ϕ. The requirement that $V(x)$ must be positive definite rather than positive semidefinite is ensured by imposing the requirement that $V(x) - \varphi(x) \geq 0$, $\forall x$ where $\varphi(x)$ is positive definite. It then follows that $V(x) - \varphi(x) \geq 0 \Rightarrow V(x) \geq \varphi(x) > 0$.

A tutorial on Lyapunov function construction using SOS methods is presented in [36], as well as non-polynomial systems analysis via SOS programming in [37,38]. In addition to stability, other properties such as region of attraction [39], delay robustness [40, 41] and controller synthesis [42, 43] have been investigated

using SOS methods. For an overview of SOS techniques in control theory see the special issue of IEEE Transactions on Automatic Control [44].

Recall from Theorem 2 that (assuming no sparsity) for a multivariate polynomial $p \in \mathcal{R}[x]$ of degree $2d$ the dimension of the monomial vector $Z(x)$ in (2.16) is $\binom{n+d}{d}$. Assume we wish to construct a Lyapunov function of degree k for the dynamical system (2.3) of a maximum degree m. It follows that $\partial\left(\frac{\partial V}{\partial x} f(x)\right) \leq m - 1 + k$, and

$$\dim\left(Z(x)\right) = \binom{n + \lceil \frac{m-1+k}{2} \rceil}{\lceil \frac{m-1+k}{2} \rceil}, \tag{2.26}$$

where $\lceil x \rceil$ denotes rounding x up to nearest integer. When n and $m + k$ are fixed (2.26) increases polynomially however when they are both allowed to vary an exponential growth in size occurs. It is this growth in complexity that makes solving large sum of squares problems intractable and provides the motivation for much of the work in this thesis.

Based on the above results it is reasonable to ask the following questions: If (2.3) is globally asymptotically stable, does there always exist an SOS Lyapunov function, moreover does there even exist a polynomial Lyapunov function? Secondly, what are the degree bounds on the Lyapunov funtion? In answer to the first question it has been shown that if (2.3) is exponentially stable on a compact set then a SOS polynomial Lyapunov function exists [45]. Furthermore Peet and Papachristodoulou [46] have shown using the Picard iteration technique that a bound on the degree of the Lyapunov function is attainable. For global stability (i.e. non-compact sets) a counterexample has been provided that shows a polynomial system for which no polynomial Lyapunov function exists [47].

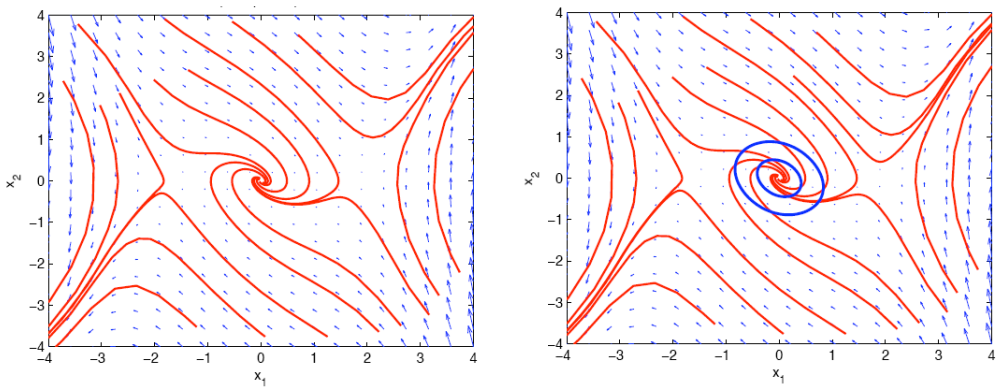
We conclude this section with a numerical example.

Example 3 *Consider the nonlinear system described the following set of nonlin-*

ear ODEs:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + \frac{1}{3}x_1^3 - x_2. \end{aligned} \tag{2.27}$$

This system has two saddle points at $(\pm\sqrt{3}, 0)$ *as well as a stable focus at the origin. Here we are interested in trajectories of (2.27) with initial conditions in the circular disc centered on the origin with radius 1:* $\mathcal{X} = \{x \in \mathcal{R}^2, \phi \in \mathcal{R} \mid x^T x - \phi \leq 0\}$ *where* $\phi = 1$*. This program was implemented in SOSTOOLS, with the result being that the equilibrium point at the origin of system (2.27) is stable inside the region* $\mathcal{X}$*. The phase portrait for the system and the level sets of the Lyapunov function are shown in Figure 2.2.*



A B

Figure 2.2: A: Trajectories of system (2.27) started form various initial conditions (red lines) can be seen to either converge to the stable equilibrium point at the origin or rapidly diverge if their initial state is close to the two saddle points at $(\pm\sqrt{3}, 0)$. B: Level curves of the Lyapunov function (blue circles).

In addition to stability, it is often desirable to determine the Region of Attraction (RoA) denoted R_A, of an equilibrium point of the unforced system (2.3). Let $\phi(t; x_0)$ be the solution to (2.3) with initial state x_0 at $t = 0$. The region of attraction corresponds to the volume of state space such that

$$R_A = \{x \in \mathcal{X} \mid \phi(t; x_0) \text{ exists and is unique, } \phi(t; x_0) \to 0 \text{ as } t \to \infty\}.$$

Determining the set R_A exactly is a nontrivial task and no analytic solutions exist for general nonlinear systems. Most region of attraction algorithms attempt to (under)approximate R_A by simple, more computationally amenable set descriptions such as ellipses and polytopes using the linear differential inclusion framework [5]. More recently, SOS methods have been applied to RoA analysis [48, 49] although these typically lead to bilinear matrix inequalities which are non-convex [50]. The most common method for approximating R_A is to determine the largest level curve of a Lyapunov function that is completely contained in $\mathcal{X}$. We will assume that $\mathcal{X}$ can be represented by a semialgebraic set of the form

$$\mathcal{X} = \{x \in \mathcal{R}^n \mid g_i(x) \leq 0, \quad i = 1, \ldots, k\}.$$

Define the point set enclosed by the level curve $V(x) = c$ where $c > 0$ by

$$\Omega_c = \{x \in \mathcal{R}^n \mid V(x) \leq c\}. \tag{2.28}$$

When $\Omega_c \subset \mathcal{X}$ and $V(x)$ is a Lyapunov function of (2.3) it follows that Ω_c is compact and positively invariant and hence a region of attraction for the zero equilibrium point of (2.3). We now propose an algorithm based on the sum of squares decomposition and the Positivstellenstaz that computes the largest level-set of V, i.e. the maximum c, such that $\Omega_c \subset \mathcal{X}$. We denote this set by Ω_{c^*}.

Given a Lyapunov function $V \in \mathcal{R}_a[x]$ and non-negative integers $\kappa, b_1, \ldots, b_k$, search for SOS polynomials $d_i \in \Sigma_{b_i}[x]$ for $i = 1, \ldots, k$ that solve:

$$\begin{aligned} \max_{d_i \in \Sigma_{b_i}[x], c>0} \quad & c \\ \text{s.t.} \quad & |x|^{2\kappa}(V(x) - c) - d_i(x)g_i(x) \in \Sigma[x], \quad i = 1, \ldots, k. \end{aligned} \tag{2.29}$$

The maximum level set is then given by $V(x) = c$.

Less conservative (although computationally more demanding) solutions can be obtained by increasing κ and the degree of the SOS polynomial multipliers b_i.

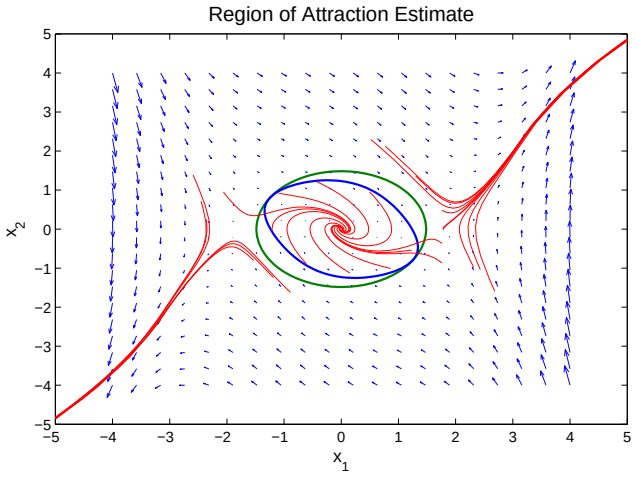


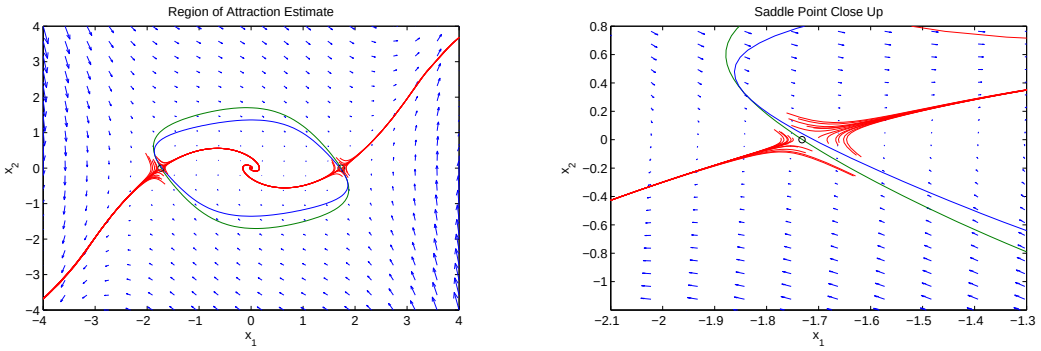
Figure 2.3: An estimate of the RoA of (2.27) obtained by solving optimization (2.29). The green curve indicates the domain $\mathcal{X}$ we are interested in, the blue curve is the largest level set of the Lyapunov function contained in $\mathcal{X}$.

Example 4 *Let us return to Example 3 involving the dynamical system given by (2.27). Consider now the domain* $\mathcal{X} = \{x \in \mathcal{R}^2 \mid x^T x - 2.2 \leq 0\}$ *shown by the green level curve in Figure 2.3. Following the steps described previously we obtain the quartic Lyapunov function:*

$$\begin{aligned} V(x) &= 3.421x_1^2 + 1.7217x_1x_2 + 2.8584x_2^2 + 0.45219x_1^4 \\ &\quad +1.318x_2x_1^3 + 1.5945x_2^2x_1^2 + 0.20294x_1x_2^3 + 0.86584x_2^4. \end{aligned}$$

Setting $\kappa = 1$ *and* $\partial(d_1) = 2$*, solving (2.29) we find that the largest level curve of* V *contained in* $\mathcal{X}$*, and thus an approximation of the RoA, is* $\{x \in \mathcal{R}^2 \mid V(x) \leq 6.308\}$ *which is shown in Figure 2.3. Improved estimates of the true region of attraction may be obtained by iterating between adjusting the domain* $\mathcal{X}$ *and then computing the largest Lyapunov level curve inside that region. The pseudo-code for such an algorithm is:*

1. *Define a domain* $\mathcal{X}$ *in the state space described by a semialgebraic set.*
2. *Construct a Lyapunov function* $V(x)$ *on* $\mathcal{X}$ *as described earlier.*
3. *Compute* Ω_{c^*} *using (2.29). Given a scalar* $\Delta > 0$*, set* $\mathcal{X}' = \Omega_{c^*+\Delta}$ *and go*



A B

Figure 2.4: A: Using the iterative method described the RoA estimate is given by the blue curve, while the green curve indicates the Domain $\mathcal{X}'$. The black circles denote the saddle point equilibria. B: Zooming in on the saddle point at $(-\sqrt{3}, 0)$ it can be seen how close our RoA estimate gets to trajectories that diverge from the stable equilibrium point.

to Step 2 replacing $\mathcal{X}$ with $\mathcal{X}'$.

The above steps should be iterated until the optimization problems become infeasible. Applying the above procedure to the example (2.27) we obtain the ROA estimate shown in Figure 2.4 (A) after 3 iterations. At this point the optimization becomes infeasible as any further increase in the the size of $\mathcal{X}'$ will include the saddle point equilibria. In Figure 2.4 (B) it can be seen how close the estimate gets to divergent, i.e. unstable trajectories.

2.5.2 SOS Scalability Issues

We now briefly summarise the main issues that prevent the use of sum of squares programming for large-scale systems analysis and provide references to work that tries to overcome these obstacles:

1. For a fixed Lyapunov function degree the dimension of the SDP that must be solved is a polynomial function of the number of states, the degree of the vector field and the degree of the Lyapunov function sought. This becomes an exponential function when the degree of the Lyapunov function is not bounded. In order to attack this problem and apply SOS techniques to more

complex problems researchers have used number theoretical techniques such as the Newton cage method [51], symmetry and group theory [52].

2. One of the long standing problems in numerical optimization is that of generating a method for solving SDPs in a decentralized manner. This is particularly challenging as SOS programmes do not produce well structured SDPs. If decentralized methods can be derived then solving very large SOS programmes will become possible. Work towards this goal is currently underway [53]. A related approach to the work carried out in Chapter 6 uses the notion of dual decomposition [54], however no method for obtaining the decomposition is provided.

3. The conservatism of using SOS programming for Lyapunov function construction is also being investigated. In particular a full characterization of the difference in volume between the cone of positive polynomials and the cone of SOS polynomials would help in determining when it is not appropriate to use SOS methods. Work along these lines can be found in [26, 55].

2.6 Conclusion

In this section we have introduced the mathematical and computational toolset that will be used throughout the rest of this work. In particular the notion of Lyapunov stability for linear and nonlinear systems has been described. Additionally it has been shown how the construction of Lyapunov functions can be formulated using semidefinite programming, i.e. as a convex optimization problem. The relationship between algebraic geometry and nonneagtive polynomials has been introduced and illustrated through numerous examples. In the remainder of this work effort will be placed on posing system analysis questions as convex optimization problems with the primary focus being placed on developing *scalable* algorithms.

Chapter 3

Model Invalidation Using SOS Programming

In this section we demonstrate how sum of squares programming can be applied to the modeling process of biological phenomena. Over the past decade the field of systems biology [2] has become one of the major success stories of interdisciplinary research. Researchers from the fields of biology, biochemistry, physics, engineering and mathematics collaborate to produce mathematical models based on observational and experimental data obtained from laboratory experiments.

The type of models used vary depending on their intended purpose, however, the most common choice of models and the type we consider in this chapter are continuous time systems of Ordinary Differential Equations (ODEs) and discrete time systems of difference equations. Other popular choices include partial differential equations (for describing spatial dynamics), functional differential equations (e.g. delay differential equations), stochastic differential equations and hybrid models that include both continuous and discrete states.

We are now at the point where the field of systems biology has reached a state where there frequently exist several *competing* models of the same biological system. Such an occurrence may create a confusion around the validity of these models and leave one asking *which is the correct model?* In reality this is an

ill-posed question as *model validation* is actually a misnomer [56]. A model can only be invalidated by experimental data or *not* invalidated. To actually validate a model would require an infinite amount of observational data. Robust control theory has been shown to provide a suitable framework for the the invalidation of biological system models [57].

In this chapter we develop different approaches for showing how competing ODE models of the same biological phenomenon containing nonlinearities and parametric uncertainty can be invalidated using experimental data. We first emphasize the strong interplay between system identification and model invalidation and we describe a method for obtaining a lower bound on the error between candidate model predictions and data. We then turn to model invalidation and formulate a methodology for discrete-time and continuous-time model invalidation. It becomes quickly evident that for even small systems the resulting SOS programmes scale poorly and become numerically ill-conditioned as the system complexity increases. One of the main contributions of this chapter is (via a reparameterization of the time variable) to improve the numerical conditioning of the invalidation problem. It is emphasized that trying to invalidate complex nonlinear models through exhaustive simulation is not only computationally intractable but also many times inconclusive.

3.1 Problem Formulation

The models we consider in this section take the following form:

$$\dot{x}(t) = f(x(t), p, t), \quad x(t) \in \mathcal{X} \quad \forall t \in [0, T] \tag{3.1}$$

where $\mathcal{X} \subseteq \mathcal{R}^n$ is a region of the state space containing the origin and $p \in \mathcal{P}$ is a vector of parameters that takes its values from a set $\mathcal{P} \subset \mathcal{R}^m$. An alternative system model which we also consider is the discrete time analogue of (3.1) given

by

$$x_{k+1} = f(x_k, p), \quad x_k \in \mathcal{X} \tag{3.2}$$

where $k \in \mathbb{Z}$ is the time index where $\mathbb{Z}$ is the set of integers and p and $\mathcal{X}$ are as defined above. We assume for both system models that the vector field f is a polynomial function of its arguments. The final assumption is not necessarily restrictive as the dynamics of biochemical reaction networks naturally take this form when the law of mass action is applied [1]. Additionally when other methods (such as Michaelis-Menten approaches) are used the resulting dynamics are governed by a rational vector field which can also be treated using the proposed framework [37].

For each of the problems considered in this chapter we require observational/experimental data

$$(t_i, \hat{x}_i), \quad i = 1, \ldots, N$$

where $\hat{x}_i \in \mathcal{X}_i \subseteq \mathcal{X}$ are the data points observed at time $t = t_i$. In the case of discrete time system models (3.2) it is assumed that $t_i \in \mathbb{Z}$. Note that the set $\mathcal{X}_i$ encodes the uncertainty in the observed data (this may be due to experimental error for example) and that $\mathcal{X}_i \subseteq \mathcal{R}^n$ implies that we *do not* require observational data for all states in the model. This assumption is particularly useful for biological models as it is often impossible to measure certain complexes. In the sections that follow we will require that all set descriptions can be written in semi-algebraic form so as to allow us to apply the concepts from Sections 2.4–2.5. Define $\hat{x}_1^i$ to be the i^{th} element of $\hat{x}$ taken at time t_1. Using this notation the uncertain observational data $\hat{x}_1^i \in \left[\underline{\hat{x}_1^i}, \overline{\hat{x}_1^i}\right]$ for $i = 1, \ldots, n$ forms a hypercube which can be represented by a finite set of polynomial inequalities:

$$\mathcal{X}_1 = \left\{ \hat{x}_i \in \mathcal{R}^n \;\middle|\; \left(\hat{x}_1^i - \underline{\hat{x}_1^i}\right)\left(\hat{x}_1^i - \overline{\hat{x}_1^i}\right) \leq 0, \quad i = 1, \ldots, n \right\} \tag{3.3}$$

With these definitions in place we address the following problems:

1. Given the observed data, can the magnitude of the maximum error between the data and an optimal model be quantified?

2. Given a continuous time ODE model (3.1) and an observed data set, is it possible to invalidate the uncertain model?

3. Given a discrete time model (3.2) and an observed data set, is it possible to invalidate the uncertain model?

In the following sections we answer these questions and stress that all solutions are arrived at without the need for simulating the proposed model. The theory and algorithms presented will be illustrated with examples from mathematical biology although all results are applicable to any system representable by continuous or discrete ODEs.

3.2 Parameter Estimation

The link between parameter estimation (referred to as system identification [58] in the control literature) and model invalidation was first highlighted in the context of control by Smith and Doyle [59]. The problem was posed in a traditional robust control Linear Fractional Transform (LFT) framework where the central question posed was: Given experimental input/output data and a nominal model, does there exist an external input and set of perturbations to the model that allows the closed loop system to reproduce the experimental data exactly? Models which failed this test could then be safely invalidated or refined. Subsequently further related problems were studied in [56, 60, 61].

The framework of the previously mentioned work exploits the linearity in candidate models and uses frequency domain techniques. However, most realistic system models will contain nonlinearities. In this work we use Lyapunov based methods which allow us to work directly with the nonlinear models. Before proceeding to the validation/invalidation problem we first illustrate how sum

of squares programming can be used for system identification. The problem addressed here is:

Given a nominal experimental data set, what is the smallest attainable error one can expect between the data and predictions obtained from an optimally fitted model subject to a well-defined parametric uncertainty?

Most system identication problems try to find an optimal parameter set in order to minimize an objective function of the error between model predictions and data. The question we are asking here is dual to that: Given an allowable parameter set, how bad will the optimal model be? If the error is large, then this could indicate that the model structure may be inappropriate and one may want to invalidate the model and repeat the system identication cycle.

Consider the continuous time ODE (3.1) with a fixed parameter vector p and initial condition $x(0) = x_0$. Picard's method of successive approximations [62] tells us that $x(t)$ is a solution of this initial value problem if and only if $x(t)$ is continuous and satisfies

$$x(t) = x_0 + \int_0^T f(x(\tau), p)d\tau. \tag{3.4}$$

Successive approximations to this equation are defined by the sequence of functions

$$\begin{aligned} u_0(t) &= x_0 \\ u_{k+1}(t) &= x_0 + \int_0^T f(u_k(\tau), p)d\tau, \quad k = 0, 1, 2, \ldots \end{aligned}$$

From this sequence it can be seen that a system identification objective function

should seek to find an optimal parameter vector $p^* \in \mathcal{P}$ such that

$$\begin{aligned} p^* &= \arg\min \sum_{i=1}^{N-1} \|\hat{x}_{i+1} - x_{i+1}\|^2 \\ &= \arg\min \sum_{i=1}^{N-1} \left\| \hat{x}_{i+1} - \hat{x}_i - \int_{t_i}^{t_{i+1}} f(x(t), p) dt \right\|^2, \end{aligned} \tag{3.5}$$

where N is the number of observed data points and x_{i+1} refers to $x(t)$ where we have assumed that the discretized time steps are aligned (in time) with the observed data point $\hat{x}_{i+1}$, i.e. $i + 1 = t$. For the case of discrete time systems the integral term is replaced by a recursive summation and the representation is exact. For continuous systems the integral term needs to be approximated as many times an analytical solution does not exist.

The optimization problem (3.5) is non-convex when the parameters in f do not enter affinely or when the observed data is uncertain. When this happens determining p^* becomes numerically challenging. Instead of solving (3.5) directly we propose a method for obtaining a lower bound on the error, i.e. find the largest $\gamma \geq 0$ such that

$$\sum_{i=1}^{N-1} \left\| \hat{x}_{i+1} - \hat{x}_i - \int_{t_i}^{t_{i+1}} f(x(t), p) dt \right\|^2 \geq \gamma. \tag{3.6}$$

In this setting γ is a lower bound on the error between the observed data and the model that has been fitted. For the case of continuous-time systems with known initial conditions we will approximate the continuous dynamics by a discrete time system via a simple Euler discretization, that is $\dot{x} = f(x, t)$, $x(0) = x_0$ is approximated by

$$x_{k+1} = x_k + \Delta f(x_k, k) \tag{3.7}$$

where Δ is the step size. We note that more sophisticated discretization methods exist with better error propagation and convergence properties, however for the purposes defining this framework (3.7) is sufficient.

The task of maximizing γ over (3.6) can be solved using Positivstellensatz arguments from Section 2.5 when γ enters affinely and the terms in (3.6) are rational. For a detailed treatment of polynomial optimization solutions using sum of squares techniques see [29, 33, 63]. To make it clear, we will now derive a general SOS programme that produces a lower bound to a polynomial optimization problem and then show how it can be used to solve (3.6).

Consider the polynomial optimization problem where $f, g \in \mathcal{R}[x]$, with $x \in \mathcal{R}^n$:

$$\min_x \quad f(x) \tag{3.8a}$$

$$\text{s.t.} \quad g(x) \leq 0 \tag{3.8b}$$

We would like to find a lower bound i.e. $\inf f(x)$ over the semi-algebraic set $g(x) \leq 0$. It should be clear that γ is a lower bound for (3.8) if $\{x \in \mathcal{R}^n \mid g(x) \leq 0,\ f(x) < \gamma\} = \emptyset$. Equivalently this can be written as

$$\{x \in \mathcal{R}^n \mid g(x) \leq 0,\ f(x) \leq \gamma,\ f(x) - \gamma \neq 0\} = \emptyset. \tag{3.9}$$

Applying Theorem 3: If there exist sum of squares polynomials $s_i \in \mathcal{R}[x]$, $i = 1, \ldots, 4$ and an integer $\kappa > 0$ then (3.9) is empty if [1]

$$s_1 - s_2(f - \gamma) - s_3 g + s_4(f - \gamma)g + (f - \gamma)^{2\kappa} = 0.$$

Substituting $\kappa = 1$ and $s_1 = s_3 = 0$ gives

$$-s_2(f - \gamma) + s_4(f - \gamma)g + (f - \gamma)^2 = 0,$$

[1] To keep the notation as readable as possible we suppress the function arguments until the final solution is presented.

dividing through by $(f - \gamma)$ we have

$$-s_2 + s_4 g + f - \gamma = 0,$$

which, after rearrangement gives

$$f(x) - \gamma + s_4(x)g(x) \in \Sigma[x]. \tag{3.10}$$

Thus by solving

$$\begin{aligned} \max_{\gamma,\ s_4 \in \Sigma[x]} \quad & \gamma \\ \text{s.t.} \quad & (3.10) \end{aligned} \tag{3.11}$$

a lower bound for (3.8) is obtained. Tighter approximations may be achievable by increasing $\partial(s_4)$. The optimization problem (3.11) can easily be adjusted to the case of multiple inequality constraints. It should also be noted that there is no assumption on convexity of the objective function or constraints. In fact, this approach is particularly useful for obtaining bounds on non-convex optimization problems.

In the case of (3.6) there are no inequality constraints imposed, thus (3.10) reduces to maximizing γ such that $f(x) - \gamma \in \Sigma[x]$. However, it is likely that $f(x)$ will be a rational function, in which case the constraints are of the form

$$f(x) \triangleq \frac{n(x)}{d(x)} > 0.$$

The corresponding optimization problem that gives a lower bound for (3.6) when f is a rational function is given by

$$\begin{aligned} \max_{\gamma} \quad & \gamma \\ \text{s.t.} \quad & n(x) - \gamma d(x) \in \Sigma[x]. \end{aligned} \tag{3.12}$$

In such optimization problems it is also desirable to obtain the argument that achieves the optimum, in this case the vector $x \in \mathcal{R}^n$ corresponding to γ. It follows that x can be computed by solving a system of equations after computing a sum of squares decomposition and performing a Cholesky decomposition $Q = B^T B$:

$$\begin{aligned} & n(x) - \gamma d(x) \in \Sigma[x] \\ & \quad \Rightarrow n(x) - \gamma d(x) &&= Z(x)^T Q Z(x) \quad (Q \succeq 0) \\ & &&= Z(x)^T B^T B Z(x) \\ & &&= h(x)^T h(x). \end{aligned}$$

Therefore, solving $h(x) \triangleq BZ(x) = 0$ for x will give the desired result.

We will now illustrate these ideas through a biochemical reaction network example.

3.2.1 Biochemical Reaction Network Example

Consider a simple biochemical reaction network with a product degradation term given by

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P, \quad P \xrightarrow{k_3} \emptyset. \tag{3.13}$$

The enzyme, E, binds reversibly with the substrate, S, to form a complex ES which then forms a product P and releases the enzyme E. The parameters on the arrows denote the rate constants and are used to quantitatively describe the speed of a reaction in a given direction. The concentration of the reactants is denoted by lowercase letters where $[\cdot]$ denotes concentration and $e = [E]$, $p = [P]$, $s = [S]$ and $c = [ES]$. We will use the *law of mass action* [1] to derive a set of dynamic equations that represent how the concentration of the species changes with time. The law states that the rate of reaction is proportional to the product of the concentrations of the reactants. Applying this to (3.13) and observing a

conservation of the free and the bound enzyme, gives:

$$\dot{s} = -k_1(e_0 - c)s + k_{-1}c, \tag{3.14a}$$

$$\dot{c} = k_1(e_0 - c)s - k_{-1}c - k_2c, \tag{3.14b}$$

$$\dot{p} = k_2c - k_3p, \tag{3.14c}$$

and $e(t)$ can be calculated from the conservation relation $e(t) = e_0 - c(t)$ where $e(t_0) = e_0$. The initial conditions are $s(t_0) = s_0$, $c(t_0) = 0$ and $p(t_0) = 0$. We will use model (3.14) to generate our experimental data that will be used to fit a model to.

The model we will try to fit to the experimental data takes the form:

$$\dot{\bar{s}} = -\frac{V\bar{s}}{K + \bar{s}} \tag{3.15a}$$

$$\dot{\bar{p}} = \frac{V\bar{s}}{K + s} - \lambda\bar{p} \tag{3.15b}$$

This type of model will be recognized by mathematical biologists as a Michaelis-Menten model [1]. There is no specific reason for selecting such a model (typically one may choose this type of model if they believe there is a clear time scale separation in the observed data), instead a generic polynomial model could have equally well been fitted.

The Euler discretization of (3.15) that we will try and fit is

$$\bar{s}_{k+1} = \bar{s}_k - \Delta\left(\frac{V\bar{s}_k}{K + \bar{s}_k}\right) \tag{3.16a}$$

$$\bar{p}_{k+1} = \bar{p}_k + \Delta\left(\frac{V\bar{s}_k}{K + \bar{s}_k} - \lambda\bar{p}_k\right). \tag{3.16b}$$

As it stands the error between the Euler discretization (3.16) parameterized by

$p = [V, K, \lambda]^T$ and N data points generated from the model (3.14) is

$$\sum_{i=1}^{N} \left\{ \left(\hat{s}_{k+1} - \bar{s}_k + \Delta \frac{V\bar{s}_k}{K + \bar{s}_k} \right)^2 + \left(\hat{p}_{k+1} - \bar{p}_k - \Delta \frac{V\bar{s}_k}{K + \bar{s}_k} + \Delta\lambda\bar{p}_k \right)^2 \right\}. \quad (3.17)$$

The error term will be biased towards fitting models with fewer observational points. To counteract this we can multiply (3.17) by a normalization factor N^{-1}. Additionally, a weighted version of (3.17) may allow for a greater error around points where there is a large transient response or to capture the most salient parts of a dynamic model as identified by a specialist.

Define the functions $d_k(K) = (K + \bar{s}_k)$ and

$$\begin{aligned} n_k(K, V, \lambda) &= [(\hat{s}_{k+1} - \bar{s}_k)(K + \bar{s}_k) + \Delta V \bar{s}_k]^2 \\ &\quad + [(\hat{p}_{k+1} - \bar{p}_k + \Delta\lambda\bar{p}_i)(K + \bar{s}_k) - \Delta V \bar{s}_k]^2 . \end{aligned}$$

Using these functions it is now possible to write (3.17) in terms of a sum of squares optimization constraint required in optimization (3.12). The constraint is now given as

$$\sum_{k=1}^{N} n_k(K, V, \lambda)^2 + \prod_{k \in \{1,\ldots,N\} \setminus i} d_k(K)^2 - \gamma \prod_{k \in \{1,\ldots,N\}} d_k(K)^2 \in \Sigma[x]. \quad (3.18)$$

The optimization objective is to maximize γ subject to (3.18) which represents the lowest bound achievable for the difference between model predictions and observed data. This can be thought of as obtaining a global lower bound to a nonlinear least squares optimization problem. Once such a value for γ has been obtained we can substitute this back into the original equation and solve for the optimal parameter vector p as described earlier.

Applying this optimization to data generated by model (3.14) we try to fit a discretized Euler model of the form (3.16). For the purpose of this example we have chosen to fit the model to 9 data points from (3.14) with parameters

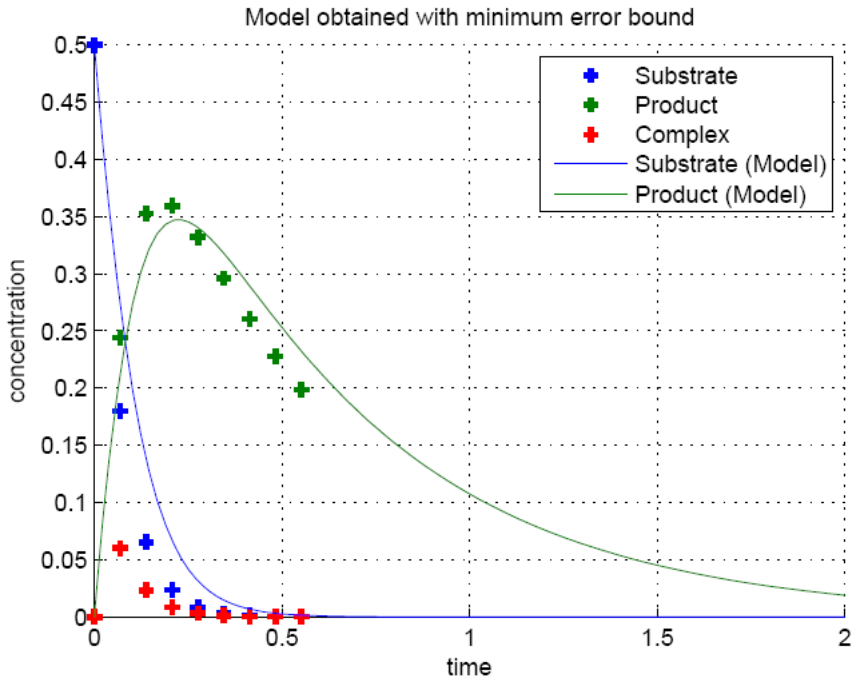


Figure 3.1: Applying optimization (3.11) to the experimental data denoted by + we obtain the optimal nominal parameter set $p = [V, K, \lambda]^T$ for the model (3.15) as denoted by the continuous blue and green lines.

$k_1 = 4.5(\text{nM s})^{-1}$, $k_{-1} = 2.5\text{s}^{-1}$, 56s^{-1}, $k_3 = 2\text{s}^{-1}$ and $e_0 = 3.5\text{nM}$. The results of the optimization problem are shown in Figure 3.1 where the observed data and the fitted model are plotted. A value of $\gamma = 0.00328$ was obtained and the fitted parameters correspond to $K = 1.313$, $V = 14.856$ and $\lambda = 1.736$. A low value for γ suggests that the model fits the data well, visually this is seen in Figure 3.1. The real benefit of having a value for γ is to use as a comparison between multiple fitted models.

The algorithm presented is very general and can easily be adjusted to handle systems with constraints and uncertainty in the observed data. It may also be possible to improve the quality of the fitted model by using higher order polynomial and SOS polynomial multipliers at the expense of computational efficiency. In the following sections a method for invalidating a biological model is presented. Typically we would pursue this option if the lower bound on the error i.e. γ is deemed to be particularly high.

3.3 Continuous Time Model Invalidation

In this section an algorithm for invalidating a continuous-time model in the form of (3.1) is presented. The methodology follows the work of Prajna [64, 65] who introduced the concept of *barrier certificates*, functions of state, parameter and time whose existence certifies that a system trajectory cannot enter a given region of the state space.

Definition 12 (Invalidation) *Suppose we are given some observational data* $\hat{x}_T \in \mathcal{X}_T \subset \mathcal{R}^n$ *and a system model of the form (3.1) with initial conditions* $x_0 \in \mathcal{X}_0 \subset \mathcal{R}^n$ *and parameters* $p \in \mathcal{P} \subset \mathcal{R}^p$*. Let* $\phi(x_0, t_0, t, p)$ *be a solution to (3.1). The model is said to be invalidated by* $\{\mathcal{X}_0, P, \mathcal{X}_T\}$ *if it can be shown that*

$$\left\{\phi(x_0, t_0, t = T, p) \bigcap \mathcal{X}_T\right\} = \emptyset, \quad \forall (x_0, x(t), p) \in \mathcal{X}_0 \times \mathcal{X} \times \mathcal{P}. \tag{3.19}$$

In words, Definition 12 states that a model is *invalid* if there does not exist a parameter choice $p \in \mathcal{P}$ such that *any* system trajectory initialized from the set $\mathcal{X}_0$ intersects with the observed data at time $t = T$.

It should be made clear that by necessity $\mathcal{X}_T \subset \mathcal{X}$. In most realistic instances the size of $\mathcal{X}$ will far outweigh the size of the measurement data sets. The definition above can easily be strengthened to include observational data taken at multiple time instances, although for ease of presentation we shall restrict ourselves to the case of uncertain initial conditions and data at just one time instance.

The following example illustrates the invalidation concept. Consider the reaction network described by (3.14). Denote the state vector $x = [[S], [C], [P], [E]]^T$, assume that all initial conditions are known to be 0 at time $t = 0$ apart from the substrate, for which we have $\hat{x}_0^1 \in \mathcal{X}_0$ where $\mathcal{X}_0 = [18, 20]$. Assume that we have

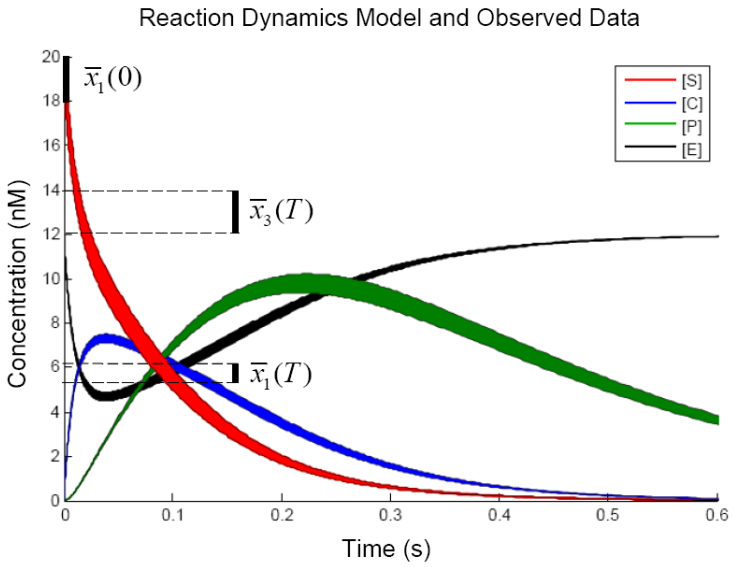


Figure 3.2: The hull of all allowable trajectories of (3.14) for an uncertain initial condition for $[S]$. The black bars denote the uncertainty in the measurements. The system trajectories do not intersect the observed data and thus the model can be invalidated.

observational data for the substrate and product taken at time $T = 0.165$s where

$$\mathcal{X}_T = \left\{(\hat{x}_T^1, \hat{x}_T^3) \in \mathcal{R}^2 \mid \hat{x}_T^1 \in [5.7, 6.2],\ \hat{x}_T^3 \in [12, 14]\right\}.$$

These sets together with the allowable trajectories and initial condition constraints are illustrated in Figure 3.2. In order to be able to illustrate all allowable trajectories we assume that there is no parametric uncertainty. This system model can be invalidated as a visual inspection shows that the system model fails to replicate the observed data. Note that in general it is not possible to invalidate a model by visual inspection.

What would be desirable is to determine exactly when a model can be invalidated, i.e. when a model of the form (3.1) cannot replicate the observed experimental data. The following Lyapunov-like theorem gives an exact answer to the invalidation problem that does *not* rely on system simulations or determining a solution, $\phi(\cdot)$ to (3.1).

Theorem 4 ([65]) *Given a model of the form (3.1) and sets $\mathcal{X}, \mathcal{X}_0, \mathcal{X}_T, \mathcal{P}$ with*

$f(x,p,t)$ being continuous in x and t. Suppose that there exists a real-valued function $B(x,p,t)$ that is differentiable with respect to x and t such that

$$\begin{aligned} B(x_T,p,T) - B(x_0,p,0) &> 0 \\ \forall (x_T,x_0,p) &\in \mathcal{X}_T \times \mathcal{X}_0 \times \mathcal{P}, \qquad (3.20) \\ \frac{\partial B}{\partial x}(x,p,t) f(x,p,t) + \frac{\partial B}{\partial t}(x,p,t) &\leq 0 \\ \forall (x,p,t) &\in \mathcal{X} \times \mathcal{P} \times [0,T]. \qquad (3.21) \end{aligned}$$

Then the model (3.1) and its associated parameter set $\mathcal{P}$ are invalidated by the tuple $\{\mathcal{X}_0, \mathcal{X}_T, T\}$. The function $B(x,p,t)$ is referred to as a barrier certificate.

In an analogous manner to Lyapunov functions which verify the stability of an equilibrium point, the existence of a barrier certificate verifies that a system model and associated parameter set are invalid. Constructing a barrier certificate can be done via convex optimization if we relax the non-negativity constraints in Theorem 4 to the existence of sum of squares polynomials. Many authors have now applied the barrier certificate methodology to various system models, see [66–68]. However, as barrier functions are functions of state, parameter and time the number of decision variables and the dimension of the LMI constraints in the SDP that must be solved to compute the sum of squares decomposition grows very quickly. Additionally, finding numerically well conditioned solutions to this problem can prove difficult. Typically high order SOS polynomial multipliers are required in practice.

The contribution of this section is a reformulation of Theorem 4 that helps improve the conditioning of the underlying SOS optimization problem. We make two observations about polynomial barrier certificates: i) The level curves of a barrier certificate typically resemble the shape of the system trajectories. ii) Relaxing the non-negativity conditions in Theorem 4 to the existence of SOS polynomials imposes that by default $B(x,p,t)$ is polynomial in state, parame-

ter and time variables. These observations seem to contradict each other and may explain the increased computational requirements when searching for barrier certificates. While it makes sense for a barrier function to be polynomial in states and parameters, a more natural choice for the dependence of B on time is $e^{-\vartheta t} \triangleq \mu$ for a fixed ϑ. This time re-parametrization will change the derivative condition (3.21) in Theorem 4. For concreteness the barrier certificate theorem with the re-parametrized time variables is given below:

Theorem 5 *Let the model (3.1) and the sets $\mathcal{X}, \mathcal{X}_0, \mathcal{X}_T, \mathcal{P}$ be given. Assume that $f(x,p,t)$ is continuous in x and t. Suppose that there exists a real valued function $B(x,p,\mu)$ that is differentiable with respect to x and t such that (3.20) holds and*

$$\begin{aligned} \frac{\partial B(x,p,\mu)}{\partial x} f(x,p,t) - \vartheta\mu \frac{\partial B(x,p,\mu)}{\partial \mu} &\leq 0 \\ \forall (x,p,\mu) &\in \mathcal{X} \times \mathcal{P} \times [\mu_{t_0}, \mu_{t_T}] \end{aligned} \tag{3.22}$$

where $\mu_{t_0} = e^{-\vartheta t_0}$ and $\mu_{t_T} = e^{-\vartheta t_T}$. Then the model and its associated parameter set $\mathcal{P}$ are invalidated by $\{\mathcal{X}_0, \mathcal{X}_T, T\}$.

Proof: The proof follows in the same manner as presented in [65]. What is left to be shown is that $\dot{B}(x,p,\mu)$ is given by (3.22) which is shown below:

$$\begin{aligned} \frac{dB(x,p,\mu)}{dt} &= \frac{\partial B(x,p,\mu)}{\partial x}\frac{dx}{dt} + \frac{\partial B(x,p,\mu)}{\partial \mu}\frac{\partial \mu}{\partial t} \\ &= \frac{\partial B(x,p,\mu)}{\partial x} f(x,p,t) - \frac{\partial B(x,p,\mu)}{\partial \mu}\vartheta e^{-\vartheta t} \\ &= \frac{\partial B(x,p,\mu)}{\partial x} f(x,p,t) - \frac{\partial B(x,p,\mu)}{\partial \mu}\vartheta\mu. \end{aligned}$$

■

While we do not attempt to attribute a precise interpretation to the meaning of the parameter ϑ it is clearly related to the time constant of the system and should be chosen appropriately.

Remark 1 *We note that there is no reason to suggest that one should be limited*

to a single value for ϑ. Indeed, many systems evolve over multiple time scales and thus the barrier certificates could be parameterized by a family of ϑ's. However it should be noted that the barrier certificates are sensitive to these time constants and thus care must be taken when assigning their values.

An algorithm for searching for barrier certificates for model invalidation using sum of squares programming and Theorem 5 is now presented. It is assumed that a model of the form (3.1) and an associated parameter set $\mathcal{P}$ has been given. We also require that some observational data at time $t = T$ have been provided in the form of (3.3). The algorithm proceeds as follows:

1. From the system model and observed data, construct the sets:

$$\begin{aligned}
\mathcal{X} &= \{x \in \mathcal{R}^n \mid g_{\mathcal{X},i}(x) \geq 0 \quad \forall i \in I_{\mathcal{X}}\} \quad \text{(state space)},\\
\mathcal{P} &= \{p \in \mathcal{R}^m \mid g_{\mathcal{P},i}(p) \geq 0 \quad \forall i \in I_{\mathcal{P}}\} \quad \text{(allowable parameter space)},\\
\mathcal{X}_0 &= \{x \in \mathcal{R}^n \mid g_{\mathcal{X}_0,i}(x_0) \geq 0 \quad \forall i \in I_{\mathcal{X}_0}\} \quad \text{(allowable initial conditions)},\\
\mathcal{X}_T &= \{\hat{x} \in \mathcal{R}^n \mid g_{\mathcal{X}_T,i}(\hat{x}_T) \geq 0 \quad \forall i \in I_{\mathcal{X}_T}\} \quad \text{(observed data)},
\end{aligned}$$

   where the g's are polynomial functions and $I_{\mathcal{X}}$ etc. refer to index sets.

2. Fix a value for ϑ and define $\mu = e^{-\vartheta t}$ and μ_{t_0}, μ_{t_T} as in Theorem 5.

3. For each of the polynomials in the sets described in Step 1 define the SOS polynomials (and fix their degree bounds):

$$\begin{aligned}
&M_{\mathcal{P},i} \in \Sigma[x_0, \hat{x}_T, p] \quad \forall i \in I_{\mathcal{P}}, && N_{\mathcal{P},i} \in \Sigma[x, p, \mu] \quad \forall i \in I_{\mathcal{P}},\\
&M_{0,i} \in \Sigma[x_0, \hat{x}_T, p] \quad \forall i \in I_{\mathcal{X}_0} && N_{0,i} \in \Sigma[x, p, \mu] \quad \forall i \in I_{\mathcal{X}_0},\\
&M_{T,i} \in \Sigma[x_0, \hat{x}_T, p] \quad \forall i \in I_{\mathcal{X}_T} && N_{T,i} \in \Sigma[x, p, \mu] \quad \forall i \in I_{\mathcal{X}_T}.
\end{aligned}$$

4. Fix the maximum degree of $B \in \mathcal{R}[x, p, \mu]$ and select a small value for $\epsilon > 0$,

such that

$$\begin{aligned} B(\hat{x}_T, p, \mu_{t_T}) &- B(x_0, p, \mu_{t_0}) - \epsilon - \sum_{i \in I_{\mathcal{P}}} M_{\mathcal{P},i}(x_0, \hat{x}_T, p) g_{\mathcal{P},i}(p) \\ &- \sum_{i \in I_{\mathcal{X}_0}} M_{0,i}(x_0, \hat{x}_T, p) g_{\mathcal{X}_0,i}(x_0) \\ &- \sum_{i \in I_{\mathcal{X}_T}} M_{T,i}(x_0, \hat{x}_T, p) g_{\mathcal{X}_T,i}(\hat{x}_T) \in \Sigma[x, \hat{x}_T, p, t], \\ -\frac{\partial B}{\partial x}(x, p, \mu) f(x, p, t) &+ \vartheta \mu \frac{\partial B}{\partial \mu}(x, p, \mu) - \sum_{i \in I_{\mathcal{P}}} N_{\mathcal{P},i}(x, p, \mu) g_{\mathcal{P}_i}(p) \\ &- \sum_{i \in I_{\mathcal{X}}} N_{\mathcal{X},i}(x, p, \mu) g_{\mathcal{X},i}(x) \\ &+ N_T(x, p, \mu)(\mu - \mu_{t_0})(\mu - \mu_{t_T}) \in \Sigma[x, p, t]. \end{aligned}$$

When the sum of squares optimization problem above is feasible the existence of a polynomial barrier certificate invalidates the model and parameter set with respect to the observational data $\mathcal{X}_T$.

3.3.1 Biochemical Reaction Network Example Revisited

The algorithm for constructing barrier functions presented in the previous section will now be illustrated. We will continue the example from Section 3.2.1.

After applying the parameter fitting optimization algorithm a model of the form (3.15) was obtained with nominal parameters $K = 1.313$, $V = 14.856$ and $\lambda = 1.736$. The trajectories for this model with initial conditions $\bar{s}(0) = 0.5$ and $\bar{p}(0) = 0$ were plotted in Figure 3.1. It will now be shown how this model can be invalidated in the presence of observed data. In order to proceed we will first generate some new experimental data, and then we will introduce both parametric uncertainty and initial condition uncertainty into the system model.

The first step towards invalidating the model is to create new experimental data.[1] The new data are obtained in the same manner as in the previous example. The model (3.14) is simulated with the same parameters as before except for

[1] Obviously in reality this data would come from a laboratory experiment.

$k_1 = 8.5(\text{nM s})^{-1}$, $k_{-1} = 2.5\text{s}^{-1}$ and $e(0) = 4.5\text{nM}$. The simulated trajectories are then discretized and an arbitrary point is chosen to invalidate the model at. In order to replicate the uncertainty in the observed data we will allow it to vary in a set.

Additionally, uncertainty will also be introduced into the model. It will be assumed that the parameter V (units [nm]/s) is uncertain but lies in the interval $V \in [13, 15]$ and that the initial condition of the substrate $s(t_0) = s_0$ is also unknown but confined to the interval $s_0 \in [0.49, 0.51]$.

The point in time that the model will be invalidated at will be $T = 0.0923\text{s}$ where it is assumed the observed data is described by the set $\mathcal{X}_T = \{\hat{\hat{p}} \in \mathcal{R}_+ \mid \hat{\hat{p}} \in [0.4, 0.5]\}$. All this data is shown graphically in Figure 3.3. The thick vertical black line corresponds to $\mathcal{X}_T$, the shaded red and blue regions correspond to the allowable trajectories of the system model given the uncertainty present.

As with the previous illustration, this example can be easily invalidated by sight, however the intention here is to show how to apply the barrier certificate methodology in conjunction with sum of squares programming.

First a barrier certificate is constructed using a sum of squares programme corresponding to Theorem 4, i.e. without the time re-parametrization. This barrier function is bounded by monomials of up to and including degree 4 in state and degree 1 in parameter and time and is given in the Appendix to this chapter.

Examining the barrier certificate we have constructed it is clear that it is quite complex in the sense that it contains many high degree monomials containing states, parameters and time. This is surprising given how clear it is (visually) that the model should be invalidated. Using the time re-parametrized method introduced in Theorem 5 we are able to find a lower order barrier certificate where the state is bounded by monomials of degree 3 when we have $\vartheta = 0.8$.

It should be pointed out that in this example it was clear at which time instant the model should be invalidated at. In general it will not be obvious

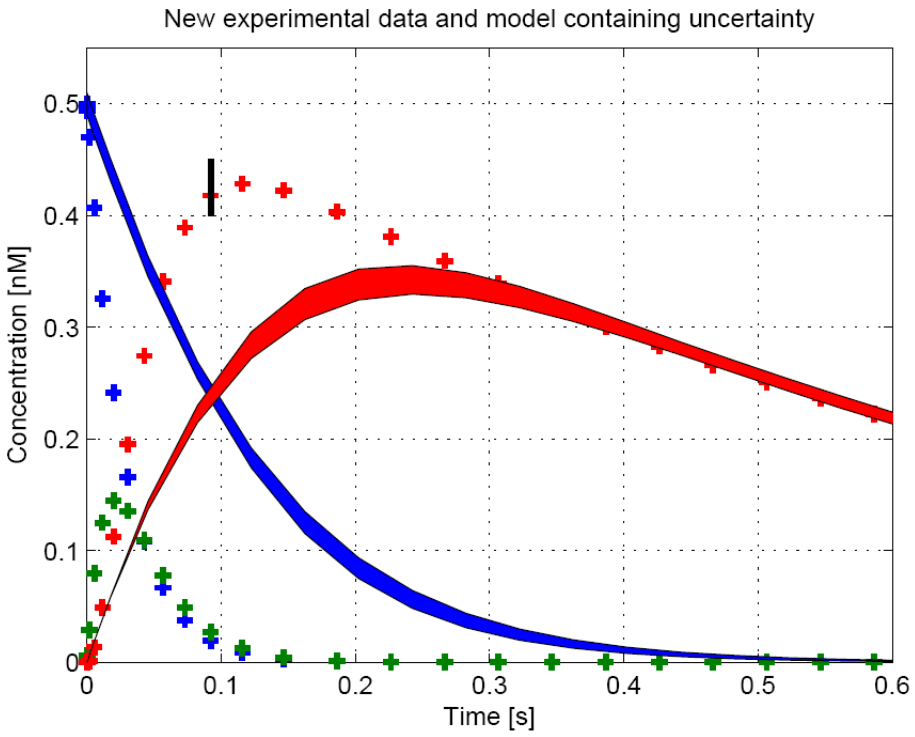


Figure 3.3: Experimental data for the complex, substrate and product shown by the red, blue and green crosses respectively. All system trajectories given the uncertainties present for the substrate (blue) and complex (red) are confined to the shaded regions. The black bar denotes the experimental data that has been selected to invalidate the model.

for cases when there are multiple uncertainties and the system models are more complex. The time point chosen in this example corresponded to the fast transient dynamics of the complex. It is left to the modeller to determine the most likely place to invalidate the model. Of course it may not be experimentally viable to obtain measurements at this time instance thus collaboration between modelers and experimentalists is required.

Ideally the invalidation process would form a part of the modelling/design cycle. What would be desirable but is still an open problem is to use information from the invalidation process can be used in order to refine the system model. There are two possibilities here; the invalidation could be used to reshape the set $\mathcal{P}$ in such a manner that invalidation is no longer possible. Alternatively, one may wish to alter the structure of the model so as to make it more difficult to be

invalidated.

Recent work that formulates a *one-shot* approach to invalidation and parameter estimation is presented in [69]. One of the drawbacks of this methodology is that it results in a non-convex optimization problem.

3.4 Discrete Time Model Invalidation

In this section a method for invalidating a discrete-time model in the form of (3.2) in the presence of observational data is presented. The approach taken is perhaps the most obvious application of the Positivstellensatz that we will encounter in this thesis. In a similar manner to the algorithms presented in the previous section we require a (possibly uncertain) candidate model and some observed data. We will only pursue the case where we wish to invalidate a model at one time instant although the algorithm generalizes to more complex cases.

Let us assume that we wish to invalidate (3.2) at time $k = T$ and that the initial condition is x_k with $k = 0$ is known. From the definition of the model we have that

$$x_T = f(x_{T-1}, p)$$

which can in turn be written as

$$x_T = f(f(x_{T-2}, p), p).$$

This recursion can be continued until we have an identity that directly relates the initial condition, x_0, to x_T, i.e. the point at which we wish to invalidate the model. It follows that

$$x_T = f(x_{T-1}, p) = f(f(x_{T-2}, p), p) = \ldots = f(f(\cdots(f(x_0, p), p), \cdots, p)$$

which we denote by

$$x_T = F(x_0, p). \tag{3.23}$$

It is assumed that (3.2) is polynomial and so it is true that (3.23) will also be polynomial. The invalidation problem can now be directly stated as a set emptiness test. In particular, if

$$\{x_0 \in \mathcal{X}_0,\ x_T = F(x_0, p),\ x_T \in \mathcal{X}_T,\ p \in \mathcal{P}\} = \emptyset \tag{3.24}$$

then given the allowable initial conditions and parameters it is not possible for the model to replicate the observed data and thus the model is said to be invalid. The task is to now apply Theorem 3 to (3.24) in order to derive a sum of squares programme for model invalidation. We begin by noting that in general the systems we are concerned with will not usually contain any inequity ($\neq$) relations, so without loss of generality we can replace h by $1 \neq 0$ so that (2.19) becomes

$$f + g = -1.$$

As was done in the continuous-time case, it is assumed that the sets $\mathcal{X}_0, \mathcal{X}_T$ and $\mathcal{P}$ are represented by semi-algebraic sets. Let g_i, $i = 1, \ldots, M$ be non-negative polynomials in $(x_0, \hat{x}_T, p)$ corresponding to the sets $\mathcal{X}_0, \mathcal{X}_T, \mathcal{P}$ where $M = \text{card}(I_{\mathcal{X}_0} \cup I_{\mathcal{X}_T} \cup I_{\mathcal{P}})$.[1] By direct application of the Positivstellensatz, if there exist SOS polynomials $s_i \in \Sigma[x_0, x_T, p]$ for $i = 1, \ldots, M$ and a polynomial $t \in \mathcal{R}[x_0, x_T, p]$ such that

$$-\left(\sum_{i=1}^{M} s_i(x_0, x_T, p) g_i(x_0, x_T, p) + t(x_0, x_T, p) h(x_0, x_T, p) + 1\right) \in \Sigma[x_0, x_T, p] \tag{3.25}$$

where $h(x_0, x_T, p) = F(x_0, p) - x_T$ then the model can be invalidated.

These ideas will be illustrated via a simple example in the following section.

[1] See Step 1 of the continuous time invalidation algorithm for the relationship between $\mathcal{X}_T$ and its polynomial representation.

3.4.1 Population Growth Example

In this section an example of discrete time model invalidation is applied to a simple population growth model. The model used to generate the observation data is generated from the *delayed logistic* equation [1]:

$$\hat{x}_{k+1} = \lambda \hat{x}_k (0.7 - 6\hat{x}_{k-1}) \tag{3.26}$$

where $\lambda \in [1.5, 2]$. The model that we wish to invalidate ignores the delay term and takes the form

$$x_{k+1} = \lambda x_k (1 - x_k) \tag{3.27}$$

where λ is as in equation (3.26). A plot of the behaviour of the two models is shown in Figure 3.4. The observational data shown corresponding to (3.26) is shown in red, while the model system (3.27) is shown in blue. Both systems are initialised from $x_0 \in [0, 0.1]$.

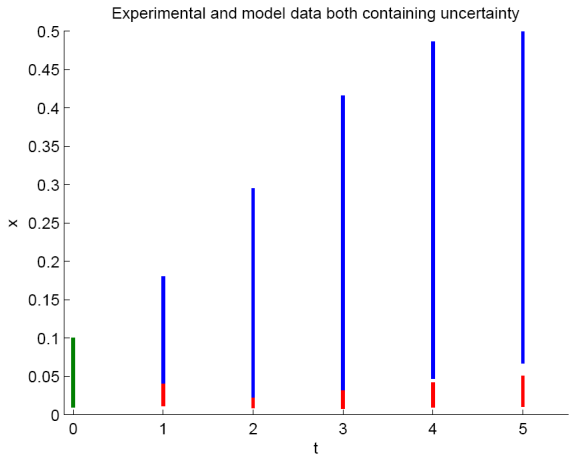


Figure 3.4: Data generated from the delayed logistic model (3.26) (red) and candidate model (3.27) (blue) for $\lambda \in [1.5, 2]$. Both models are initialised from $x_0 \in [0, 0.1]$ illustrated by the green bar.

This example has been chosen as it can easily be verified visually that the model is invalid. Invalidation corresponds to the blue and red bars in Figure 3.4 not overlapping. While it is obvious that the model can be invalidated at $k = 4$

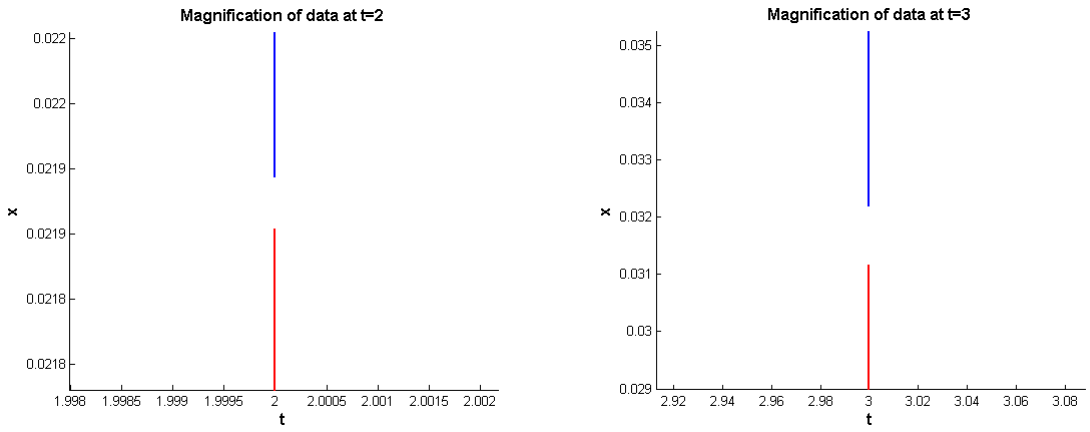


Figure 3.5: Closer inspection shows that the model fails to replicate the observed data at $k = 2$ (left) and $k = 3$ (right).

or $k = 5$, further examination shows that the model can also be invalidated at $k = 2$ and $k = 3$ as shown in Figure 3.5.

We will proceed to invalidate (3.27) at $k = 2$ where $\hat{x}_2 \in [0.00225, 0.016384] \triangleq [\underline{\hat{x}_2}, \overline{\hat{x}_2}]$ (red bars in Figure 3.4). The following sets are required in order to proceed:

$$\begin{aligned}
\mathcal{P} &= \{\lambda \in \mathcal{R} \mid -((\lambda - 1.5)(\lambda - 2)) \geq 0\}, \\
\mathcal{X}_0 &= \{x_0 \in \mathcal{R} \mid -((x_0 - 0.01)(x_0 - 0.1)) \geq 0\}, \\
\mathcal{X}_2 &= \{\hat{x}_2 \in \mathcal{R} \mid ((\hat{x}_2 - 0.00225)(\hat{x}_2 - 0.016384)) \geq 0\}.
\end{aligned}$$

This leads to the following set emptiness requirement:

$$\left\{ \begin{array}{cc} g_1 = -((\lambda - 1.5)(\lambda - 2)) \geq 0, & g_2 = -((x_0 - 0.01)(x_0 - 0.1)) \geq 0 \\ g_3 = -((\hat{x}_2 - 0.00225)(\hat{x}_2 - 0.016384)) \geq 0 & \hat{x}_2 - F_2(x_0, \lambda) = 0 \end{array} \right\} = \emptyset \tag{3.28}$$

where

$$F_2(x_0, \lambda) = \lambda^2 x_0(1 - x_0)(1 - \lambda x_0(1 - x_0)) \triangleq h.$$

The sum of squares programme of the form (3.25) that verifies that the set (3.28) is empty corresponds to finding a polynomial $t \in \mathcal{R}[x_0, x_T, \lambda]$ and sum of squares

polynomials $s_i \in \Sigma[x_0, \hat{x}_2, \lambda]$ for $i = 1, 2, 3$ such that

$$-(1 + s_1 g_1 + s_2 g_2 + s_3 g_3 + th) \in \Sigma[x_0, \hat{x}_2, \lambda].$$

For this example, setting $\partial(s_1) = \partial(s_2) = \partial(s_3) = 2$ and $\partial(t) = 1$ the model is successfully invalidated with

$$\begin{aligned}
t &= 1734.1 - 453.27\lambda \\
s_1 &= 19869\hat{x}_2^2 + 19373x_0^2 + 100.290\lambda\hat{x}_2 + 95.810\lambda x_0 + 44.915\lambda^2 \\
s_2 &= 18705 + 924.30\hat{x}_2 + 17074\hat{x}_2^2 + 2995x_0 - 177.783x_0\hat{x}_2 + 27973x_0^2 \\
&\quad -8481.2\lambda + 207.98\lambda\hat{x}_2 - 5423.4\lambda x_0 + 8388.8\lambda^2 \\
s_3 &= 18807\hat{x}_2^2 - 70.162x_0\hat{x}_2 + 17071x_0^2 + 889.04\lambda\hat{x}_2 - 104.154\lambda x_0 \\
&\quad +21490\lambda^2.
\end{aligned}$$

The coefficients of the polynomials above are large, indicating that the infeasibility certificate is very sensitive to the model structure and parameters. This is to be expected as in this case the model comes very close to replicating the observed data. The computational effort required to invalidate the model at later time instances increases as the polynomial $F(x_0, p)$ (3.23) increases in degree. In fact it is this increase in complexity that is the limiting factor of the proposed approach. What would be desirable is an efficient method for propagating the uncertain state forward in time. The result of such a scheme would mean that (3.23) would not need to relate the data to the initial conditions but instead to a later time point thus reducing the degree of (3.23). One possible method for achieving this uses the concept of *tubes*, a recently developed technique from the field of robust model predictive control [70]. However, in this example at $k = 3$ and $k = 4$ the coefficients in the polynomials t and s_i drop by orders of magnitude as the invalidation becomes easier.

In a similar manner to the continuous-time invalidation problem, choosing the

point in time to invalidate more complex models requires understanding of the model and collaboration from the data provider.

3.5 Conclusion

In this chapter the versatility of sum of squares programming and its relevance to problems arising in mathematical biology has been demonstrated. It has been shown how the SOS programming framework can be used for i) parameter estimation, ii) continuous-time model invalidation and iii) discrete-time model invalidation.

In the first instance, it was shown that parameter estimates for nonlinear models could be obtained and a bound on the smallest possible error between model and data is achieved. Next, by re-paremetrizing the barrier certificate theorem introduced in [65] we showed how computational savings could be achieved when invalidating nonlinear models containing uncertainty. Finally an algorithm based directly on the Positivstellensatz was described that can invalidate uncertain discrete time models. The discrete time algorithms could potentially be extended by introducing a parameterized time variable as was done with the continuous time case. Additionally ideas from tube MPC could be used to reduce the degree of the final sum of squares decomposition that must must be performed.

3.6 Appendix

The barrier certificate obtained for the biochemical reaction network in Section 3.3.1:

$$
\begin{aligned}
B(x,V,t) =& 0.0107x_1x_2t + 0.372x_2V + 0.798\times10^{-3}x_1Vt + 0.231\times10^{-2}tx_1^4 \\
&+ 0.218\times10^{-2}tx_2^3x_1 + 0.482\times10^{-3}x_2Vt + 0.322\times10^{-4}tx_2^4x_1 \\
&+ 0.521\times10^{-2}tx_1^3 + 0.578\times10^{-4}tx_1^3x_2^2 + 0.132x_1^5 + 0.0918x_2^5 \\
&+ 0.654\times10^{-3}x_1tVx_2^2 + 0.330x_2x_1^2 - 0.643\times10^{-4}x_2^3tV - 0.241\times10^{-2}x_2^2x_1t \\
&+ 0.564\times10^{-4}x_1tVx_2^3 + 0.0170Vx_1^2 + 0.0151x_2^2x_1^2 - 0.249\times10^{-2}tx_2Vx_1 \\
&+ 0.408Vx_2^2 + 0.635x_1x_2^2 - 0.555Vx_1^3 - 0.203x_2x_1^3 - 0.663\times10^{-3}tVx_2^2 \\
&- 0.551\times10^{-3}tx_2^3 - 0.536\times10^{-3}tx_1^4x_2 + 0.443Vx_1^3x_2 + 0.197\times10^{-3}tx_2x_1^2V \\
&+ 0.0628Vx_2^3 + 0.459\times10^{-3}tx_2^4 + 0.0618x_1x_2^3 - 0.114\times10^{-3}tx_1^5 \\
&+ 0.177\times10^{-3}tx_2^3x_1^2 + 0.0187x_1Vx_2^2 - 0.969\times10^{-2}x_1Vx_2^3 \\
&+ 0.769Vx_2^2x_1^2 + 1.30x_1^2Vx2 - 0.584x_1Vx_2 + 0.654x_2x_1^4 + 0.459x_1x_2^4 \\
&+ 0.076x_1 + 0.238x_2 - 0.246\times10^{-2}Vx2^4 + 0.274\times10^{-2}x_1^2t + 0.859x_2^2x_1^3 \\
&+ 0.800x_2^3x_1^2 - 0.005x_2t + 0.996\times10^{-4}tVx_1^3 - 0.522x_1x_2 + 0.402\times10^{-2}tx_2x_1^3 \\
&- 0.157\times10^{-4}tx_2^4V + 0.877\times10^{-2}tx_2x_1^2 - 0.728\times10^{-3}tVx_1^2 \\
&+ 0.387\times10^{-3}tx_1^2Vx_2^2 + 0.414x_1V + 0.317\times10^{-2}tx_2^2x_1^2 - 0.185\times10^{-2}x_2^2t \\
&- 1.43x_2^2 + 0.949Vx_1^4 - 0.258x_1^2 - 0.0117x_1t - 0.326\times10^{-4}tx_1^4V \\
&- 0.447\times10^{-5}x_1^5V + 0.166x_1^3 - 0.0362x_1^4 + 0.382x_2^3 + 0.0928x_2^4 \\
&+ 0.251\times10^{-3}tVx_2x_1^3 + 0.646\times10^{-5}tx_2^5.
\end{aligned}
$$

Chapter 4

System Decomposition via Composite Lyapunov Functions

In this chapter we investigate methods for analyzing large-scale dynamical systems that are of sufficient complexity that traditional *direct* methods such as those presented in the previous chapters cannot be applied. We focus on the case of constructing Lyapunov functions for dynamical systems of the form $\dot{x} = f(x)$ where the state vector $x \in \mathcal{R}^n$ is sufficiently large that computing sum of squares decompositions becomes computationally intractable.

Two methods for reducing the complexity in this type of analysis are proposed. The first method is a systems-level approach that aims to *decompose* the system of interest into multiple subsystems of reduced state dimension, connected in feedback. Such an approach allows for a subsystem-by-subsystem Lyapunov analysis to be performed with the ultimate goal of verifying stability of the original large-scale system through the stability properties of its decomposed subsystems. The second method is numerical in nature and can be used in conjunction with the composite Lyapanov framework described above. The idea here is to maximize the sparsity of the monomials present in a sum of squares Lyapunov function.

The chapter is organised as follows: In Section 4.1 the problem of applying SOS methods to large-scale systems is described and the system decomposition

methodology is formulated. The necessary background material covering algebraic graph theory and composite Lyapunov functions is presented in Section 4.2. A method for constructing sparse polynomial Lyapunov functions is presented in Section 4.3 and the main decomposition algorithm is given in Section 4.4. The ideas presented are illustrated with numerical examples throughout the chapter.

4.1 Problem Formulation

Consider the nonlinear system given by

$$\dot{\hat{x}}(t) = \hat{f}(\hat{x}(t)) \tag{4.1}$$

where $\hat{x}(t) \in \mathcal{R}^n$ and $\hat{x}(0) = \hat{x}_0$ and define the set $\mathcal{T} = \{\hat{x}_1, \ldots, \hat{x}_n\}$. For the sake of brevity we will omit the time argument and refer to $\hat{x}(t)$ as $\hat{x}$. Without loss of generality assume that the equilibrium point of interest is at the origin i.e. $f(0) = 0$; and assume that the vector field $\hat{f} : \mathcal{X} \to \mathcal{R}^n$ is polynomial, and $\mathcal{X} \subseteq \mathcal{R}^n$ is a domain that contains 0. The aim of this work is to describe a framework for determining if the origin of (4.1) is asymptotically stable when n and/or $\partial(\hat{f})$ are too large to make a direct sum of squares based analysis tractable. Currently, this typically occurs when both $n > 6$, $\partial(\hat{f}_i) \geq 4 \ \forall i$ and there is non-trivial coupling in the dynamics.

Our solution to scaling SOS based Lyapunov function construction is to *decompose* system (4.1) into N subsystems:

$$\begin{aligned}
\dot{x}_1 &= f_1(x_1) + g_1(x_1, u_1), \qquad y_1 = x_1, \qquad u_1 = [y_2^T, \ldots, y_N^T]^T && \text{(4.2a)}\\
&\vdots \qquad\quad \vdots \qquad\qquad\qquad \vdots \qquad\qquad \vdots \\
\dot{x}_N &= f_N(x_N) + g_N(x_N, u_N), \quad y_N = x_N, \quad u_N = [y_1, \ldots, y_{N-1}^T]^T && \text{(4.2b)}
\end{aligned}$$

where $g_i(x_i, 0) = 0$ for $i = 1, \ldots, N$ and all x_i and the elements of $\hat{x}$ have been permuted to produce a new state vector $x = [x_1^T, \ldots, x_N^T]^T$ with $x_i \in \mathcal{R}^{n_i}$, and $\sum_{i=1}^{N} n_i = n$.

Construct the sets $\mathcal{T}_i = \{x_i^1, \ldots, x_i^{n_i}\}$ (where x_i^j denotes the j^{th} element of x_i) and note that $\bigcup_{i=1}^{N} \mathcal{T}_i = \mathcal{T}$ and impose the restriction that $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$ for all $i, j = 1, \ldots, N$, $i \neq j$. This final requirement states that the decomposition must produce disjoint subsystems. Given the state partitions $\mathcal{T}_i$ we denote the autonomous dynamical subsystem corresponding the states in $\mathcal{T}_i$ by $S(\mathcal{T}_i)$.

In Section 4.4.3 we will relax the restriction of disjoint partitions and allow for overlapping partitions when one of the subsystems is unstable.

Definition 13 (Subsystem Lyapunov function) *Given the dynamical system $\dot{x} = f(x) + g(x, u)$, if there exists a continuously differentiable function $V : \mathcal{R}^n \to \mathcal{R}$ such that $V(x) > 0 \ \forall x \neq 0$, $V(0) = 0$ and $\frac{\partial V}{\partial x} f(x) < 0 \ \forall x \neq 0$ then $V(x)$ is called a subsystem Lyapunov function.*

The objective of system decomposition is to find a partition as described in (4.2) in a manner that allows us to verify stability of the overall system (4.1) from the subsystem Lyapunov functions of (4.2a–4.2b): we note that this is a combinatorial problem and that an exhaustive search becomes computationally intractable for large n.

4.1.1 Motivation

The typical assumption made when dealing with large-scale systems is that they consist of a number of smaller interconnected subsystems. *Composite* methods proceed by analyzing each subsystem in turn and then make conclusions about the large-scale connected system based on the information obtained from the subsystem analysis. Typically, bounds are placed on the allowable strength of interaction between subsystems. Stability of the connected system is then verified through a composite Lyapunov function (see Section 4.2.2) which takes the form

of a weighted sum of subsystem Lyapunov functions. Bailey [71] first proposed such an approach in order to apply Lyapunov theory to interconnected linear systems. Input-output properties of nonlinear multi-input multi-output systems were studied side by side in [72] where the main results were derived using the theory of M-matrices, leading naturally to the concept of diagonal Lyapunov functions and stability in the presence of perturbations [73]. Moylan and Hill [74] also treated input-output and Lyapunov stability in parallel using the theory of dissipative systems [75]. Their results were then generalized to include finite-gain, passive and conic systems. More recently the concept of V-stability [76] was introduced where the same Lyapunov function is used to cover each of the subsystems in the full system.

The critical assumption made above is that the system of interest is an aggregation of smaller subsystems. Such an assumption is many times restrictive. For example, models obtained through system identification techniques or integrative system models are not likely to exhibit modular structure in their dynamics.

The main result of this chapter is a method for *decomposing* nonlinear dynamical systems into a collection of lower-order interacting subsystems in such a manner that facilitates a composite analysis. Such an approach is desirable as many times, large-scale dynamical systems do not present themselves in 'modular' form and at the same time, it may happen that an apparent modularity in the system may not permit analysis using composite Lyapunov functions. Note also that, implicit in the composite stability framework is the assumption that each subsystem must be stable. It will be shown later that by allowing for non-disjoint decompositions (i.e. allowing states to be members of more than one subsystem) it is possible to verify the stability of systems that originally consisted of unstable subsystems.

We observe that the idea of decomposing a dynamical system has also been proposed in [77, 78] although there is no description of how to obtain such a decomposition. The closest approach to that which is presented here is by Callier

et al [79] where a graph theoretical decomposition is obtained by searching for all strongly connected components in the system graph. In comparison to this and previous work we present an algorithmic approach (that avoids an exhaustive search) for deriving the interacting subsystems and show that such a decomposition facilitates the search for a composite stability certificate. We then couple the decomposition approach with SOS programming techniques for nonlinear system analysis. Within this decomposition/SOS framework, we provide an additional sparsity-based approach for further computational improvement.

4.2 Preliminaries

Many large-scale dynamical systems are formed by an aggregation of smaller subsystems, these systems are referred to as networked systems. Typically networked systems are represented by graphs. *Algebraic graph theory* provides a matrix framework for representing the topology of such systems. Here we present some of the most important aspects from algebraic graph theory. For a concise treatment of the subject the reader is referred to [80] and [81]. In addition to topology, we are concerned with stability. The stability of networked systems has been of interest to researchers in the control community since the pioneering work of Bailey [71] in the mid-60s. Here we present a technique based on composite Lyapunov functions that will be used throughout this chapter.

4.2.1 Algebraic Graph Theory

Define an *undirected weighted* graph as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{Z})$ where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N vertices or nodes, $\mathcal{E} = \{e_1, \ldots, e_M\} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge-set and $\mathcal{Z} = \{z_1, \ldots, z_M\}$ where $z_j > 0$ is the weight of edge j. Associated with $\mathcal{G}$ is a symmetric weighted adjacency matrix $\mathcal{A}(\mathcal{G}) \in \mathcal{R}^{N \times N}$ where $[\mathcal{A}(\mathcal{G})]_{ij} > 0$ if there exists an edge connecting v_i to v_j. For *unweighted* graphs, it is assumed that $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, which corresponds to $\mathcal{Z} = \{1\}$, in which case the

graph will be denoted $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The weighted $N \times N$ Laplacian matrix associated with a graph is defined by

$$\mathcal{L}(\mathcal{G}) = \text{diag}(\mathcal{A}(\mathcal{G})\mathbf{1}) - \mathcal{A}(\mathcal{G}).$$

The Laplacian matrix has several interesting properties: for undirected graphs $\mathcal{L}(\mathcal{G}) \succeq 0$ and the number of zero eigenvalues is equal to the number of connected components in the graph. The smallest non-zero eigenvalue of $\mathcal{L}(\mathcal{G})$ is called the *Fiedler eigenvalue* and is denoted $\lambda_F(\mathcal{L}(\mathcal{G}))$. We will make frequent use of the spectral properties of graphs [35] in the sequel.

The incidence matrix, $\mathcal{C}(\mathcal{G}) \in \mathcal{R}^{N \times M}$ of an undirected graph is defined by assigning an arbitrary direction to each $e_i \in \mathcal{E}$ and setting

$$[\mathcal{C}(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } e_i \text{ enters } v_j \\ -1 & \text{if } e_i \text{ leaves } v_j \\ 0 & \text{otherwise} \end{cases}.$$

The weighted Laplacian can then be equivalently defined using the incidence matrix by

$$\mathcal{L} = \mathcal{C}(\mathcal{G})\mathcal{W}(\mathcal{G})\mathcal{C}(\mathcal{G})^T$$

where $\mathcal{W}(\mathcal{G}) \in \mathcal{R}^{M \times M}$ is a diagonal matrix and $[\mathcal{W}(\mathcal{G})]_{ii} = z_i$ with $z_i \in \mathcal{Z}$.

When it is clear from the context, we will omit the graph argument and simply refer to the adjacency, incidence, weighting and Laplacian matrices by $\mathcal{A}, \mathcal{C}, \mathcal{W}$ and $\mathcal{L}$ respectively.

4.2.1.1 Graph Partitioning

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{Z})$ the partitioning problem requires one to construct T subgraphs, $\mathcal{G}_k(\mathcal{V}_k, \mathcal{E}_k, \mathcal{Z}_k), k = 1, \ldots, T$ such that $\bigcup_{k=1}^{T} \mathcal{V}_k = \mathcal{V}$ and $\mathcal{V}_k \bigcap \mathcal{V}_l = \emptyset$ for all $k \neq l$ and $\mathcal{E}_k = \{(v_i, v_j) \in \mathcal{E} | v_i, v_j \in \mathcal{V}_k\}$ where the objective

is to minimize the sum of the weights of the edges connecting nodes in different partitions. The cut-set $\mathcal{E}_c$ is the set of edges that connect vertices from different subgraphs. The optimal cut-set is the cut-set with the minimum sum of edge weights. For the case where the number of vertices in each partition is specified the problem is known to be $\mathcal{NP}$-complete [82].

Partitioning a graph into two subgraphs (bisection) can be formulated as the following combinatorial optimization problem:

$$\begin{aligned} \min_{z} \quad & \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{A}_{ij}(\mathcal{G})(1-z_i z_j) = \frac{1}{2}z^T\mathcal{L}(\mathcal{G})z \\ \text{s.t.} \quad & z^T\mathbf{1} \neq \pm n, \\ & z_i^2 = 1 \quad i = 1,\ldots,n. \end{aligned} \tag{4.3}$$

Here $z_i = 1$ if node v_i is in one partition and $z_i = -1$ if it is in the other. The spectral partitioning algorithm [83] approximates (4.3) by dropping the constraints. Each vertex is then assigned to a partition according to the rule $z_i = \text{sign}(\mathbf{y}_i)$ where $\mathbf{y}_i$ is the i^{th} element of the eigenvector corresponding to $\lambda_F(\mathcal{L}(\mathcal{G}))$. For fixed size partitions one can sort $\mathbf{y}$ into ascending order and form a partition at the median. In order to obtain multiple partitions the algorithm is recursively called on the subgraphs produced by the previous decomposition.

Alternative convex approximations that use semidefinite programming for combinatorial problems are presented in [18, 84]. In addition, an SOS based approach using the Positivstellensatz was presented in Section 2.5.

4.2.2 Composite Lyapunov Functions

In Chapter 2 an algorithmic method for constructing Lyapunov functions for polynomial systems was described. An upper bound on the computational complexity (in terms of the dimension of the LMI associated with the underlying SDP) was also given (2.26). For large n it becomes intractable to solve the SDP

unless the system has a very specific structure that can be used.

Frequently, it is possible to model a system as an interconnection of lower order subsystems of the form

$$\dot{x}_i = f_i(x_i) + g_i(x), \quad i = 1, \ldots, N \tag{4.4}$$

where the state vector of the i^{th} subsystem is $x_i \in \mathcal{R}^{n_i}$, $x = [x_1^T, \ldots, x_N^T]^T$ and $\sum_{i=1}^{N} n_i = n$. It is implicitly assumed that the vector fields f_i, g_i, $\forall i$ are sufficiently smooth so as to ensure local existence and uniqueness of all solutions with initial conditions in the domain of interest. We follow the treatment of composite Lyapunov functions presented by Khalil [4].

Composite methods consist of two steps: i) First the stability of each of the subsystems of (4.4) is analyzed in isolation, i.e. the N autonomous systems:

$$\dot{x}_i = f_i(x_i). \tag{4.5}$$

ii) The next step is to combine the results of the previous step with information of the interconnection dynamics, $g_i(x)$ in order to verify stability of (4.4).

Assume that for each of the N subsystems (4.5) a Lyapunov funcion $V_i(x_i)$ exists that satisfies the requirements of Theorem 1. In the sequel we will use SOS programming (2.25a–2.25b) to construct the Lyapunov functions. The function

$$V_c(x) = \sum_{i=1}^{N} \alpha_i V_i(x_i) \quad \alpha_i > 0 \tag{4.6}$$

is called a composite Lyapunov function (CoLF) for the isolated subsystems. It is clear that $V_c(x) > 0$ for all $x \neq 0$ for any choice of $\alpha_i > 0$. However, $V_c(x)$ may not be a valid Lyapunov function for (4.4) as $\dot{V}_c(x)$ is not necessarily negative definite along the system trajectories as the interconnection dynamics were ignored. The

CoLF derivative along the system trajectories is

$$\dot{V}_c(x) = \sum_{i=1}^{M} \alpha_i \frac{\partial V_i}{\partial x_i} f_i(x_i) + \sum_{i=1}^{M} \alpha_i \frac{\partial V_i}{\partial x_i} g_i(x). \tag{4.7}$$

By construction the left hand term of (4.7) is negative definite, the right hand term however will in general be indefinite. Assume that the subsystem Lyapunov functions, $V_i(x_i)$ are quadratic as described in Section 2.3.1[1] and assume that the subsystem Lyapunov functions satisfy

$$\begin{aligned} \frac{\partial V_i}{\partial x_i} f_i(x_i) &\leq -\kappa_i \varphi_i^2(x_i) \\ \left\| \frac{\partial V_i}{\partial x_i} \right\| &\leq \upsilon_i \varphi_i(x_i) \end{aligned}$$

for $i = 1, \ldots, N$ and $\|x\| < d$ for some positive scalars κ_i, υ_i and for some positive definite continuous functions $\varphi_i : \mathcal{R}^{n_i} \to \mathcal{R}$. If the interconnection dynamics satisfy the bounds

$$\|g_i(x)\| \leq \sum_{j=1}^{N} \gamma_{ij} \varphi_j(x_j), \quad \gamma_{ij} \geq 0, \quad \text{for } i = 1, \ldots, N \tag{4.8}$$

then the derivative of the CoLF $V_c(x) = \sum_{i=1}^{N} \alpha_i V_i(x_i)$ along the trajectories of (4.4) satisfies the inequality

$$\dot{V}_c(x) \leq -\frac{1}{2} \varphi^T (DS + S^T D) \varphi$$

where $\varphi = [\varphi_1(x_i), \ldots, \varphi_N(x_N)]^T$, $D = \text{diag}(\alpha_1, \ldots, \alpha_N)$ and $S \in \mathcal{R}^{N \times N}$ with elements

$$S_{ij} = \begin{cases} \kappa_i - \upsilon_i \gamma_{ii} & \text{if } i = j \\ -\upsilon_i \gamma_{ij} & \text{if } i \neq j \end{cases}.$$

The following statement summarizes the results of this section:

[1] Such an assumption is not necessary, however it simplifies the remaining discussion.

Corollary 1 *Given the dynamical system (4.4), if there exists a diagonal matrix $D \succ 0$ such that $DS + S^T D \succ 0$ then the derivative of the CoLF (4.6) is negative definite along the system trajectories which verifies the stability of (4.4). Systems which satisfy this property are said to be diagonally stable.*

The matrix S in the above discussion has a special structure and is known as an M-matrix or *Metzler* matrix which we formally define below. There is a strong connection between stability of interconnected systems, diagonal stability and M-matrices [85] as we shall show later in Section 4.4.

Definition 14 (Metzler Matrix) *A matrix $A \in \mathcal{R}^{n\times n}$ is said to be Metzler if and only if $[A]_{ij} \geq 0$ for all $i \neq j$.*

Some important facts about M-matrices and stability that will be used later are given in the theorem below:

Theorem 6 ([86]) *Let A be an M-matrix. Then the following statements are equivalent:*

1. *A is Hurwitz.*
2. *There exists a diagonal matrix $P \succ 0$ such that $A^T P + PA \prec 0$ (Diagonal stability).*
3. *AX has a strictly negative diagonal entry for all non-zero positive semidefinite matrices X.*

The M-matrix stability criterion can be interpreted as the requirement that strength of the subsystem interconnections must be weaker than the strength of the stability of the isolated subsystems. More recently, Langbort *et al* [87, 88] have shown how to utilize diagonally stable systems in order to convexify the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ decentralized controller synthesis problem.

A useful property of systems with dynamics governed by M-matrices that is applicable to models in systems biology is that for any initial condition taken in

the positive orthant, the system flow, e^{At}, $t \geq 0$ will remain in that orthant i.e. if A is an M-matrix then

$$\dot{x} = Ax, \quad x_0 \in \mathcal{R}^n_+ \quad \Rightarrow \quad x(t) \in \mathcal{R}^n_+ \quad \forall t \geq 0. \tag{4.9}$$

This invariance property is typically used to construct structured Lyapunov functions for large-scale networked systems such as the population dynamics model we present later in this chapter.

4.3 Sparse Lyapunov Functions

In this section a method for reducing the number of decision variables in the SDPs associated with constructing a CoLF is presented. Reducing the complexity of these SDPs for SOS programming has begun to receive some attention. In [52,89] algebraic properties such as symmetry are exploited to reduce the number of decision variables present. Pre- and post-processing methods are described in [90] to improve numerical stability of the decomposition and in [51] the *Newton cage*[1] method for bounding the dimension of the monomial vector $Z(x)$ for sparse polynomials is presented.

The method presented here is tailored to the case of composite Lyapunov function construction and can be used in conjunction with the methods referred to above. Consider the dynamical system $\dot{x} = f(x)$ and assume that we are searching for a polynomial Lyapunov function $V(x)$. Examining (2.26) it is clear that the dimension of the LMI constraints is a function of:

1. The number of variables in the polynomial $p(x)$, i.e. $\dim(x)$.

2. The degree of the vector field $f(x)$, i.e. $\partial(f)$.

3. The degree of the Lyapunov function sought, i.e. $\partial(V)$.

[1] Alternatively this is referred to as the Newton polytope.

Observe that this complexity is specific to the Lyapunov derivative inequality. Using composite methods the positivity of the N subsystem Lyapunov functions,V_i, decouples into N problems of dimension $\binom{n+d}{d}$ where $\partial(p(x)) = 2d$ and $x \in \mathcal{R}^n$. The majority of the computational burden lies with the derivative condition. Recall that n_i denotes the dimension of the i^{th} subsystem. Let $\widehat{Z}(x)$ denote the vector of monomials (2.16) associated with a composite SOS Lyapunov function. Then (2.26) is reduced to

$$\dim(\widehat{Z}(x)) = \sum_{i=1}^{N} \binom{n_i + \lceil \frac{\partial(f)-1+\partial(V_i)}{2} \rceil}{\lceil \frac{\partial(f)-1+\partial(V_i)}{2} \rceil}. \tag{4.10}$$

Each of the subsystem Lyapunov functions is computed independently, however we still need to check that the derivative of the CoLF along the full system trajectories is a sum of squares, i.e. that there exist scalars $\alpha_i > 0$ for $i = 1, \ldots, N$ such that

$$-\frac{\partial \sum_{i=1}^{N} \alpha_i V_i(x_i)}{\partial x} f(x) - \varphi(x) \in \Sigma[x]. \tag{4.11}$$

The method for reducing the dimension of $\widehat{Z}(x)$ (4.10) and in turn (4.11) that we propose is to make the subsystem Lyapunov functions as sparse (in the sense of the number of monomials) as possible.

In the context of a polynomial $p \in \mathcal{R}[x]$ with K monomials, let θ_i refer to the i^{th} monomial coefficient and let $\Theta = [\theta_1, \ldots, \theta_K]^T$. A sparse polynomial is a polynomial where the number of non-zero elements in Θ is small. Sparse polynomials of degree $2d$ require fewer than all $\binom{n+d}{d}$ monomials in $Z(x)$ which in turn decreases the dimension of the underlying LMI.

Maximizing sparsity of a vector is a non-convex optimization problem [6]. An alternative heuristic is to minimize the ℓ_1-norm of a vector:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \quad x \in \mathcal{R}^n. \tag{4.12}$$

The following SOS programme can be used to construct a sparse Lyapunov

function: Given a constant $\epsilon > 0$, search for a polynomial $V(x)$ with associated vector of coefficients Θ and vector of monomials $W(x)$ and solve

$$\min_{\beta_i \geq 0, V \in \mathcal{R}[x]} \quad \sum_{i=1}^{K} \beta_i \tag{4.13a}$$

$$\text{s.t.} \quad -\beta_i \leq \theta_i \leq \beta_i, \quad \beta_i \geq 0 \quad i = 1, \ldots, K \tag{4.13b}$$

$$V(x) = \Theta^T W(x) \tag{4.13c}$$

$$V(x) - \epsilon x^T x \in \Sigma[x] \tag{4.13d}$$

$$-\frac{\partial V}{\partial x} f(x) - \epsilon x^T x \in \Sigma[x]. \tag{4.13e}$$

In the above optimization, (4.13a-4.13b) encode the ℓ_1-norm minimization. Constraints (4.13d-4.13e) enforce the Lyapunov inequalities. The added constraints will increase the computation time by a small margin when constructing Lyapunov functions, however, the time taken to verify composite Lyapunov functions can be significantly reduced as we illustrate with examples at the end of the chapter.

For large systems the difference between the dimensions of $Z(x)$ and $\widehat{Z}(x)$ can mean the difference between tractable and intractable analysis. For $k = 3$ and $m = 4$, a 75% reduction can be obtained for bisections (N=2); for trisection (N=3) this becomes 89%. This sparse Lyapunov function algorithm will be incorporated into the system decomposition algorithm presented in the following section.

4.4 Decomposition

We now present an algorithm for decomposing the dynamical system (4.1) into N subsystems of the form (4.2) connected in feedback. For simplicity we will present the algorithm for the case of $N = 2$.[1] Recall that our goal is to obtain 2

[1] To obtain multiple subsystems one can simply recursively call the decomposition algorithm on the subsystems obtained in the pervious iteration.

subsystems of the form

$$\left\{\begin{array}{lcl}\dot{x}_1 &=& f_1(x_1)+g_1(x_1,u_1)\\ y_1 &=& x_1\\ u_1 &=& y_2\end{array}\right. , \quad \left\{\begin{array}{lcl}\dot{x}_1 &=& f_2(x_2)+g_2(x_2,u_2)\\ y_2 &=& x_2\\ u_2 &=& y_1\end{array}\right. , \tag{4.14}$$

such that a composite Lyapunov function formed from the isolated subsystems

$$\dot{x}_1 = f_1(x_1), \quad \dot{x}_2 = f_2(x_2)$$

verifies the stability of the equilibrium point at the origin of (4.1).

Before proceeding it will be useful to formally define the group of permutation matrices as we shall make liberal use of them throughout the rest of this thesis.

Definition 15 (Permutation Matrix) *The matrix $T \in \mathcal{R}^{n\times n}$ is said to be a permutation matrix if and only if*

1. *T is a binary matrix, i.e. $[T]_{i,j} \in \{0,1\}\ \forall(i,j)$.*
2. *Each row of T sums to one.*
3. *Each column of T sums to one.*

The group of $n \times n$ permutation matrices is denoted Π_n.

A useful property of permutation matrices is that for $T \in \Pi_n$ we have $T^T = T^{-1}$.

Consider a Linear Time Invariant (LTI) system of the form

$$\dot{\hat{x}} = \widehat{F}\hat{x} \tag{4.15}$$

with $\hat{x} \in \mathcal{R}^n$ and $\widehat{F} \in \mathcal{R}^{n\times n}$. For $T \in \Pi_n$, let

$$\begin{aligned} F &= T\widehat{F}T^{-1} \\ x &= T\hat{x} \end{aligned}$$

which is a permutation of the system matrix and state vector. A decomposition (bisection) of this LTI system gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} \tag{4.16}$$

with the matrices $F_{ii} \in \mathcal{R}^{n_i \times n_i}$ and $F_{ij} \in \mathcal{R}^{n_i \times n_j}$. Recall that (4.16) is stable if given a matrix $Q \succ 0$ there exists a $P \succ 0$ satisfying $F^T P + PF + Q = 0$. Further, P is unique and $V(x) = x^T P x$ is a Lyapunov function for (4.16).

An equivalent formulation of (4.16) is

$$\dot{x}_1 = F_{11}x_1 + u_1, \quad y_1 = F_{21}x_1, \quad u_1 = y_2, \tag{4.17a}$$

$$\dot{x}_2 = F_{22}x_2 + u_2, \quad y_2 = F_{12}x_2, \quad u_2 = y_1. \tag{4.17b}$$

A desirable decomposition will turn (4.15) into (4.17) such that a block diagonal Lyapunov function of the form

$$V_c(x) = x^T \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix} x, \quad P_{ii} \in \mathcal{S}_{++}^{n_i}, \quad i = 1, 2 \tag{4.18}$$

for (4.16) exists. Note that (4.18) can equivalently be written as

$$V_c(x) = \alpha_1 x_1^T P_{11} x_1 + \alpha_2 x_2^T P_{22} x_2, \quad \alpha_1, \alpha_2 > 0.$$

Work related to this problem has appeared in [91] where quadratic Lyapunov functions with a specific structure or *subspace constraint* are searched for using convex optimization. Surprisingly however, they do not present any result for the block diagonal case of (4.18). In fact there does not appear to be many widely known results for the block diagonal case, although the authors of [91] comment:

We mention that a structured Lyapunov stability problem with block-diagonal structure is briefly mentioned in Khalil and Kokotovic [92].

The following theorem determines when a block diagonal Lyapunov function exists.

Theorem 7 *Consider (4.15) with $\widehat{F}$ Hurwitz decomposed into (4.17) where F_{11} and F_{22} are Hurwitz. A block diagonal Lyapunov function of the form $V_c(x) = \alpha_1 x_1^T P_{11} x_1 + \alpha_2 x_2^T P_{22} x_2$ for some $\alpha_i > 0$ exists for (4.15) if and only if*

$$\begin{bmatrix} \alpha_1 Q_1 & -\alpha_1 P_{11} F_{12} - \alpha_2 F_{21}^T P_{22} \\ -\alpha_1 F_{12}^T P_{11} - \alpha_2 P_{22} F_{21} & \alpha_2 Q_2 \end{bmatrix} \succ 0, \tag{4.19}$$

where $P_{ii} \succ 0$ solve the Lyapunov equations $F_{ii}^T P_{ii} + P_{ii} F_{ii} + Q_i = 0$ for $i = 1, 2$ and Q_i given.

Proof: The derivative condition for diagonal Lyapunov function is $\dot{V}_c(x) = \alpha_1 \dot{x}_1^T P_{11} x_1 + \alpha_1 x_1^T P_{11} \dot{x}_1 + \alpha_2 \dot{x}_2^T P_{22} x_2 + \alpha_2 x_2^T P_{22} \dot{x}_2$. Substituting the decomposition (4.16) and $F_{ii}^T P_{ii} + P_{ii} F_{ii} = -Q_i$ into $\dot{V}_c$ we obtain (4.19). ∎

Proposition 2 *Consider system (4.15) with $\widehat{F}$ Hurwitz, which has been decomposed according to (4.17a–4.17b) with $n_1 = n_2$ so that both subsystems are strictly passive and are interconnected in negative feedback. Then there always exists an α for which (4.18) is a Lyapunov function for (4.15).*

Proof: The strict passivity assumption for the two subsystems ensures that $P_{11} F_{12} = -I$ and $P_{22} F_{21} = I$ and $Q_i \succ 0$ where the negative feedback interconnection has been imposed by negating F_{12}. Assuming $\alpha_1 = 1$ and taking the Schur complement of condition (4.19) gives

$$\alpha_2 Q_1 \succ (\alpha_2 F_{21}^T P_{22} + P_{11} F_{12}) Q_2^{-1} (\alpha_2 P_{22} F_{21} + F_{12}^T P_{11})$$

This, after substitution reads:

$$Q_1 Q_2 \succ \frac{1}{\alpha}(\alpha - 1)^2 I$$

Hence given Q_1, Q_2, an appropriate value for α_2 can always be found as $\frac{1}{\alpha_2}(\alpha_2 - 1)^2 \in [0, \infty]$ for $\alpha_2 > 0$. ■

The following theorem based on block diagonal dominance of LMI (4.19) gives sufficient conditions for (4.19) to be satisfied thus ensuring that a diagonal Lyapunov function exists.

Theorem 8 *When* $\|\alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}\| < min\{\alpha_1 \underline{\sigma}(Q_1), \alpha_2 \underline{\sigma}(Q_2)\}$ *a block diagonal Lyapunov function for (4.15) exists if* Q_1, Q_2 *are* M*-matrices.*

Proof: A partitioned matrix $A = \begin{bmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{bmatrix}$ where $A_{ii} \in \mathcal{R}^{n_i \times n_i}$ and $A \in \mathcal{R}^{n \times n}$ is said to be *strictly block diagonally dominant* if the submatrices on the diagonal of A, A_{ii} are nonsingular and

$$(\|A_{ii}^{-1}\|)^{-1} > \sum_{k=1, k \neq i}^{M} \|A_{ik}\| \quad \text{for } i = 1, \ldots, M,$$

where $(\|A_{ii}^{-1}\|)^{-1} = \inf_{x \neq 0} \frac{\|A_{ii} x\|}{\|x\|} = \underline{\sigma}(A_{ii})$. If A is strictly block diagonally dominant and all A_{ii} are M-matrices then the eigenvalues λ_i of A satisfy $Re(\lambda_i) > 0$ for $i = 1, \ldots, M$ [93]. LMI (4.19) has the structure $\begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix}$ with X, Z M-matrices. To ensure (4.19) is positive definite we require $(\|X^{-1}\|)^{-1} > \|Y\|$ and $(\|Z^{-1}\|)^{-1} > \|Y\|$, hence $\|Y\| < \min\{(\|X^{-1}\|)^{-1}, (\|Z^{-1}\|)^{-1}\}$. Applying this to (4.19) completes the proof. ■

While Theorem 7 gives necessary and sufficient conditions for the existence of a block diagonal Lyapunov function, Theorem 8 provides only *sufficient* conditions. It is likely that the constraint of Q being an M-matrix will be overly restrictive.

An observation from Theorems 7–8 is that in general minimizing the off diagonal blocks $\|\alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}\|$ will facilitate in satisfying the block diagonal Lyapunov function requirements.

The following theorem provides an intuitive method for choosing the matrices Q_i in Theorems 7–8 and relates this to minimizing the output energy ($\|y\|_{\mathcal{L}_2}$) of a dynamical system.

Theorem 9 *For $Q_1 = F_{21}^T F_{21}$ and $Q_2 = F_{12}^T F_{12}$, minimizing max $\|y_1\|_{\mathcal{L}_2}^2$ and max $\|y_2\|_{\mathcal{L}_2}^2$ for $u_1 = u_2 = 0$ and $\|x_i(0)\| = 1$ in (4.17a–4.17b), minimizes an upper-bound for $\|\alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}\|$.*

Proof: Denote $\Delta = \alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}$, then;

$$\begin{aligned} \|\Delta\| &= \|\alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}\| \\ &\leq \alpha_1 \|P_{11} F_{12}\| + \alpha_2 \|F_{21}^T P_{22}\| \\ &\leq \alpha_1 \|P_{11}\| \|F_{12}\| + \alpha_2 \|F_{21}\| \|P_{22}\|. \end{aligned} \tag{4.20}$$

Setting $Q_1 = F_{21}^T F_{21}$, $Q_2 = F_{12}^T F_{12}$, noting that $y_1 = F_{21} x_1$, $y_2 = F_{12} x_2$, set $\alpha_1 = \alpha_2 = 1$, and neglecting input dynamics we obtain two Lyapunov equations corresponding to the observability gramians, $F_{ii}^T P_{ii} + P_{ii} F_{ii} + Q_i = 0$, $i = 1, 2$ of (4.16). The $\mathcal{L}_2$-norm of the output of each subsystem (which we will call energy) is then given by $\|y_i\|_{\mathcal{L}_2}^2 = x_i(0)^T P_{ii} x_i(0)$. For unit initial conditions $\max \|y_i\|_{\mathcal{L}_2}^2 = \lambda_{\max}(P_{ii}) = \overline{\sigma}(P_{ii})$ where P_{ii} is unique and depends upon the choice of output mapping F_{ij}.

Consider now the observability gramian $F_{11}^T P_{11} + P_{11} F_{11} = -F_{21}^T F_{21}$. We have

$$\begin{aligned} \|F_{21}^T F_{21}\| = \|F_{11}^T P_{11} + P_{11} F_{11}\| &\leq \|F_{11}^T P_{11}\| + \|P_{11} F_{11}\| \\ &= 2\|F_{11}^T P_{11}\| \\ &\leq 2\|F_{11}\| \|P_{11}\| \\ &\Rightarrow 2\|P_{11}\| \geq \|F_{11}\|^{-1} \|F_{21}\|^2. \end{aligned}$$

Therefore, if max $\|y_1\|_{\mathcal{L}_2}^2$ is minimized so is the upperbound on $\|F_{11}\|^{-1}\|F_{21}\|^2$. Using this and an identical argument for $\|P_{22}\|$ we see that minimizing $\|y_i\|_{\mathcal{L}_2}^2$ minimizes the individual terms in (4.20) which is in turn an upperbound for $\|\Delta\|$. ■

Lemma 1 ([94]) *If A, B, C, D are bounded linear operators on a complex Hilbert space then*

$$\left\|\begin{bmatrix} A & 0 \\ 0 & D \end{bmatrix}\right\| \leq \left\|\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right\| \quad \textit{and} \quad \left\|\begin{bmatrix} 0 & B \\ C & 0 \end{bmatrix}\right\| \leq \left\|\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right\|.$$

Remark 2 *It should be observed that in Theorem 9 a particular choice of F_{ij} giving a small $\|F_{ij}\|$ may produce an increase in $\|F_{ji}\|$ and vice-versa. However from Lemma 1 it follows that the magnitude of $\|F_{ij}\|$ is bounded from above by $\|F\|$.*

Determining when a decoupled decomposition will produce a valid composite Lyapunov function is not trivial as in general one cannot assume that the sub-matrices F_{ii} are Hurwitz and iterating through all state permutations is not computationally tractable. Remarks on the complexity of optimizing over the manifold of permutation matrices that go beyond the combinatorial nature of the problem are addressed in Section 4.5.

Thus far it has been shown that in order to obtain a block diagonal composite Lyapunov function that ignores the subsystem interactions it is desirable to minimize the strength of the interaction which is encapsulated by $\|\alpha_1 P_{11} F_{12} + \alpha_2 F_{21}^T P_{22}\|$. What is important to note and not intuitively obvious is that a decomposition algorithm should make both $\|P_{ii}\|$ *and* $\|F_{ij}\|$ small, i.e. the dynamics and the coupling strength should be accounted for. It is not difficult to construct examples where a decomposition based solely on the topology fails to find a CoLF, yet a CoLF does exist for other decompositions.

4.4.1 Graphical Representation of a Dynamical System

The decomposition algorithm we propose is based on representing the dynamical system as a weighted graph. In this setting, vertices represent states of the system and weighted edges represent the strength of interconnection between states. We will make the notion of strength precise in Section 4.4.1.2. With the system represented in this manner a graph partitioning algorithm is used to cut the graph where the interconnection strength is weakest. The interpretation here is that this makes the $\|g_i(x)\|$ terms in (4.8) small. Once the partition has been performed the appropriate subsystems (4.2a–4.2b) can be constructed and a composite analysis performed. An outline of the algorithm is given below:

1. Represent system (4.1) as a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{Z})$.
2. Apply a graph bisection algorithm (such as the spectral method from Section 4.2.1.1) on $\mathcal{G}$ to obtain $\{\mathcal{G}_1, \mathcal{G}_2\}$.
3. From the subgraphs $\mathcal{G}_i$ construct the subsystems (4.2a–4.2b).
4. Search for a composite Lyapunov function using either SDP or SOS methods.

If the state dimension N is large and $\hat{f}$ is nonlinear multiple subsystems may be obtained by recursively applying Step 2 on the subgraphs. If Lyapunov functions are being constructed using SOS programming then further computational improvements can be obtained by applying the Sparse Lyapunov function algorithm presented in Section 4.3 at Step 3.

We proceed to describe how a dynamical system can be represented as a static graph which corresponds to Step 1 above.

4.4.1.1 Topology

For linear systems in the form of (4.15) the topology is defined by the directed adjacency matrix

$$[\mathcal{A}(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } \widehat{F}_{ij}^T \neq 0 \text{ and } i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

From $\mathcal{A}(\mathcal{G})$ the edge set $\mathcal{E}$ can be derived. For nonlinear systems of the form (4.1) a Jacobian linearization about the equilibrium point x^* is performed such that $\widehat{F} = \frac{\partial f}{\partial x}\big|_{x=x^*}$ and $\mathcal{A}(\mathcal{G})$ is constructed using (4.21).

We stress that the linearization is only used to construct the graphs and establish a picture of local state interaction strength: all subsequent analysis is carried out on the nonlinear system.

4.4.1.2 Edge weights

In Theorem 9 it was shown that minimizing the output energy of a system (which we defined as the $\mathcal{L}_2$-norm of the output signal) reduces the norm of the interconnection dynamics, which in turn increases the likelihood that a CoLF for the system will exist. What would be desirable in a system decomposition is to obtain states in a subsystem that interact more strongly with states in the same subsystem than states belonging to the subsystems they are connected to. In statistical physics this measure of *connectedness* or *modularity* [95] of static graphs (not dynamical systems) has received a great deal of attention and *community detection* algorithms such as those presented in [96, 97] attempt to find such clusters of vertices. At the heart of these algorithms typically lies a graph partitioning algorithm.

It was stated earlier that a system decomposition should partition the state vector (equivalently the system graph) so as to minimize the energy flow between nodes (states) in different subgraphs (subsystems). Specifically we are interested

in determining the maximum energy that can flow between nodes (states). The maximum output energy of a stable autonomous LTI system of the form

$$\dot{x} = Fx, \quad y = Cx \tag{4.22}$$

with the initial condition $x(0) = x_0$ subject to $\|x_0\| = 1$ is given by

$$\|y\|_{\mathcal{L}_2}^2 = \bar{x}_0^T P \bar{x}_0 \tag{4.23}$$

where $P \succ 0$ solves the Lyapunov equation

$$F^T P + PF + C^T C = 0 \tag{4.24}$$

and $\bar{x}_0$ is the unit vector aligned with the eigenvector corresponding to the largest eigenvalue of P (see for example [17]). However, the linearized system we are interested in (4.15) does not have an output map. This actually turns out to be advantageous as if we append the output map $y = \mathcal{C}^T(\mathcal{G})$ then the output becomes the M-dimensional column vector with entries given by $(x_i - x_j)$, for all $(v_i, v_j) \in \mathcal{E}$ where $\mathcal{E}$ is the edge set belonging to the graph defined by (4.21). This vector gives the difference between all combinations of states that interact with each other directly.

In principle it may be possible to compute nonlinear grammians [98, 99] and thus avoid the need for linearization altogether. In general computing nonlinear grammians is not possible for large systems, furthermore in many cases they do not even exist thus we have to make do with the linearizations, the caveat being that the further from the equilibrium point the system is likely to be initialized from the less accurate the edge weight will be.

A weighted adjacency matrix $\mathcal{W}(\mathcal{G}) \in \mathcal{R}^{n \times n}$ is constructed where the energy

flow from node i to node j is given by

$$[\mathcal{W}(\mathcal{G})]_{ij} = \begin{cases} \bar{x}_0^T X^{(ij)} \bar{x}_0 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

Here $X^{(ij)} \succeq 0$ solves the Lyapunov equation (4.24)

$$F^T X^{(ij)} + X^{(ij)} F = -\delta_k \delta_k^T$$

where δ_k is the k^{th} column of $\mathcal{C}(\mathcal{G})$ corresponding to edge (i,j). In order to apply the spectral partitioning algorithm the directionality of $\mathcal{W}(\mathcal{G})$ is removed by symmeterization:

$$\mathcal{W}^*(\mathcal{G}) = \frac{1}{2}(\mathcal{W}(\mathcal{G}) + \mathcal{W}^T(\mathcal{G})). \tag{4.25}$$

The weighted system graph $\mathcal{G}$ is then fully described by the set of edges and vertices $\mathcal{V}$ and $\mathcal{E}$ obtained from (4.21) and the edge weights $\mathcal{Z}$ obtained from the non-zero entries of (4.25).

4.4.2 Decomposition Algorithm

In order to encode both the fact that we need to cut where $\|F_{ij}\|$ is small and where $\|P_{ii}\|$ is small, we apply a weighted graph partitioning algorithm, where the edges are weighted by the maximum energy that can flow on them as described earlier. Combining graph partitioning with worst case energy maximization gives

rise to the following min-max problem:

$$\begin{aligned} \min_{z} \quad & \max_{x_0} \ \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}(1-z_iz_j)\left[\mathcal{A}(\mathcal{G})\right]_{ij}\int_0^\infty \left(x_i(\tau)-x_j(\tau)\right)^2 d\tau \\ \text{s.t.} \quad & \dot{x} = Fx, \quad x(0)=x_0, \\ & \|x_0\|_2^2 = 1, \\ & z_i^2 = 1 \quad i=1,\ldots,n, \\ & z^T\mathbf{1} \neq \pm n. \end{aligned} \tag{4.26}$$

where $\mathcal{A}(\mathcal{G})$ is the binary adjacency matrix that defines the state interactions. Optimization (4.26) encapsulates the dual objective of finding the maximum amount of energy (specified in terms of the $\mathcal{L}_2$ norm of the output signal) that can flow through the network and the state partition that minimizes the energy flow between the two subsystems.

The dynamical system decomposition is obtained by solving:

$$\underset{z_i^2=1}{\text{minimize}} \quad \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n}(1-z_iz_j)\,\mathcal{W}^*_{ij}(\mathcal{G}). \tag{4.27}$$

where the decision vector z defines the graph (and hence) state partition $\mathcal{X}_i$ as described in Section 4.1. Optimization (4.27) is exactly equivalent to the graph partitioning problem in Section 4.2.1.1 which can be solved using the spectral algorithm described earlier. The full decomposition algorithm is summarized below:

1. Linearize (4.1) to obtain F and construct the system graph $\mathcal{A}(\mathcal{G})$.
2. Construct $\mathcal{C}(\mathcal{G})$ and compute $\bar{x}_0$.
3. Construct the symmeterized energy matrix $\mathcal{W}^*(\mathcal{G})$.
4. Compute the Laplacian $\mathcal{L}(\mathcal{G})$ from $\mathcal{W}^*(\mathcal{G})$ and apply the spectral partitioning algorithm to $\mathcal{L}(\mathcal{G})$.

5. Use the partitioning vector, z, to construct the nonlinear subsystems (4.2).

Remark 3 *The method described above for computing edge weights requires a linearization of (4.1) in order to solve $m+1$ observability gramians. It is possible to avoid this step by computing* empirical gramians *[100] with simulation or experimental data. Such an approach can also be applied if the system under study is unstable or a finite time horizon solution is desirable.*

We now present two examples that illustrate the system decomposition algorithm. The systems chosen are typical of large-scale systems one would wish to analyze. However, neither system satisfies the typical assumptions that allow it to be interrogated in a decentralized, distributed or scalable manner.

An Internet network with users having non-concave utility functions

Analysis of network congestion control algorithms for the Internet is a well-studied problem. Most of the associated theory concentrates on cases in which the users have Utility functions that are strictly concave, non-decreasing and continuously differentiable. These assumptions simplify greatly the analysis. By employing Lyapunov arguments the global stability of the equilibrium of the nonlinear system describing the network dynamics can be analyzed for arbitrarily sized topologies and link capacities [101].

In this example we consider a network that involves users that have Utility functions that are not strictly concave. This can result in networks with multiple equilibria. For illustrative purposes we consider a *primal* congestion control algorithm [102]. For such a network with N users and L links, the *Routing* matrix $R \in \mathcal{R}^{L\times N}$ is given by:

$$R_{li} = \begin{cases} 1 & \text{if user } i \text{ uses link } l, \\ 0 & \text{otherwise.} \end{cases}$$

Link l is characterized by its capacity c_l, and what needs to be ensured is that the total flow through every link in the network does not exceed its given capacity. In vector form this is written as

$$y = Rx \leq c$$

where x is the vector of user transmission rates and y is a vector of aggregate transmission rates on the links. The aim of the congestion control algorithm is to allocate the available bandwidth on the links to competing users.

The nonlinear system describing the evolution of a user's transmission rate is:

$$\dot{x}_i = k_i(x_i - q_i(\alpha_i + x_i^2)) \tag{4.28}$$

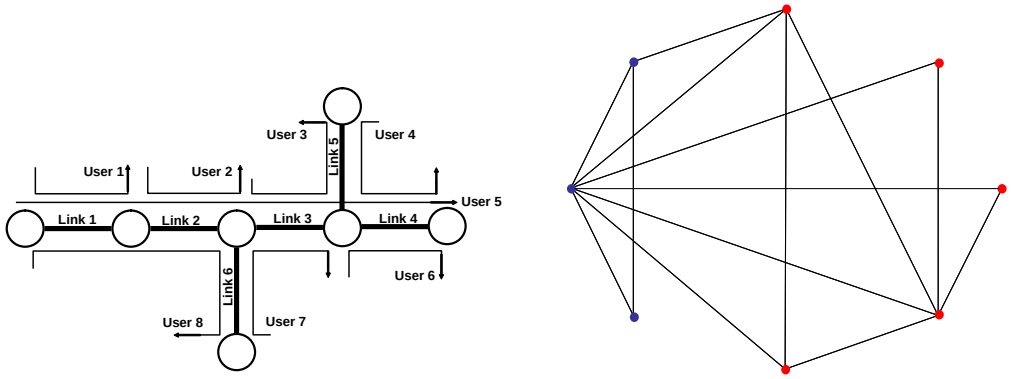
where $q_i = \sum_{l=1}^{L} R_{li} p_l$ represents the aggregate price for all the links utilized by user i and $k_i > 0$ and $\alpha_i > 0$. The Utility function for these users can be computed to be

$$U_i(x_i) = \frac{1}{2}\log(\alpha_i + x_i^2), \quad i = 1, \ldots, N$$

which are *not* concave functions and could result in problems such as the existence of multiple equilibria etc. We set $p_l = \left(\frac{y_l}{c_l}\right)$ to be the price mechanism which satisfies the conditions that p_l is an increasing function in y_l and decreasing in c_l.[1]

We analyze a system with $N = 8$ users and $L = 6$ links connected according

[1]The motivation for selecting such a pricing mechanism is as follows: When $c_l > 0$ and $y_l < c_l$ the function $p_l : \mathcal{R} \to [0, 1]$ gives the steady state overflow probability for an M/M/1 queue with arrival rate y_l and service rate c_l [102]. In addition, when a solving the underlying network utility maximization problem using a penalty function method this class of pricing mechanism ensures that the solution obtained is optimal.



A B

Figure 4.1: A: The Internet topology given by the routing matrix R. B: The interaction graph partitioned into two subsystems, shown by red and blue nodes.

to the following routing matrix

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Graphically this network is depicted in Figure 4.1. This system has a stable equilibrium in the positive orthant, as observed from simulation and linear analysis.

The 8-state nonlinear system is now analyzed by decomposing it into two smaller subsystems using the spectral decomposition method described previously. The resulting decomposition partitions the states into two sets

$$\mathcal{T}_1 = \{x_1, x_2, x_3, x_4, x_7, x_8\}, \quad \mathcal{T}_2 = \{x_5, x_6, x_7\}$$

whose interconnections are shown in Figure 4.1(B). The decomposition changes if we consider randomly weighted graphs, hence underlying the importance of using an appropriately weighted graph in the decomposition.

Two 4^{th} order Lyapunov functions, $V_1(x_1)$ and $V_2(x_2)$ were constructed for

each subsystem. We then solved another SOS programme of the form (4.11) to obtain $\alpha = 1.8544$ so that $V(x) = V_1(x_1) + \alpha V_2(x_2)$ is a Lyapunov function for the full system, thus proving that the equilibrium of the whole networked system is locally asymptotically stable. It was found that this was the only decomposition for which a 4^{th} order composite Lyapunov function could be found. We should mention here that we were unable to compute using direct SOS methods (i.e. no decomposition) a numerically reliable 4^{th} order Lyapunov function.

Lotka-Volterra Dynamics

Given a network described by Lotka-Volterra dynamics [103], it is a known result that the stability of the equilibrium in the positive orthant can be verified by a *structured* Lyapunov function if and only if the network is dissipative [104]. In this example we specifically seek a system which fails to meet this criterion and is thus computationally more difficult to analyze. Furthermore the state dimension of the system in this example is well beyond that which is currently tractable using existing SDP solvers and SOS programming.

Consider the Lotka-Volterra system that describes the population dynamics of a community,

$$\dot{x}_i = x_i \left(b_i - x_i - \sum_{j=1}^{n} \mathcal{A}_{ij}(\mathcal{G}) x_j \right), \quad x_0 \in \mathcal{R}_+^{16} \tag{4.29}$$

where x_i denotes population of species i, $b_i \in \mathcal{R}_+^{16}$ is the birth rate of species i and $\mathcal{A}(\mathcal{G}) \in \mathcal{R}^{16\times 16}$ defines the interaction structure shown in Figure 4.2 (left). The equilibrium point of interest for (4.29) is $x^* = (I + \mathcal{A})^{-1} b$.

It is not possible to construct a Lyapunov function using SOS methods directly as the state dimension is too large. Applying the decomposition algorithm the

following state partitions are produced:

$$\mathcal{T}_1 = \{x_1, x_3, x_7, x_8, x_9\}, \quad \mathcal{T}_2 = \{x_2, x_6, x_{10}, x_{15}, x_{16}\},$$
$$\mathcal{T}_3 = \{x_4, x_5, x_{11}, x_{12}, x_{13}, x_{14}\}$$

as shown in Figure 4.2 (middle, right). SOS programming is then used to construct the subsystem Lyapunov functions for $S(\mathcal{T}_i)$, for $i = 1, 2, 3$ and a composite Lyapunov function is found and verified by computing an SOS decomposition (see Section 2.4) of the derivative condition. The computational cost of computing the required Lyapunov functions is shown in Table 4.1. The Lyapunov function corresponding to $S(\mathcal{T}_i)$ is V_i and are given by:

$$\begin{aligned}
V_1 \;=\;& 0.559x_8^2 + 0.343x_8x_7 + 0.711x_7^2 + 0.0435x_8x_1 + 0.00619x_1x_7 + 0.725x_1^2 \\
&+0.0222x_8x_3 - 0.239x_7x_3 + 0.238x_1x_3 + 0.153x_3^2 + 0.0741x_9x_8 + 0.0331x_9x_7 \\
&+0.0774x_9x_1 + 0.214x_9x_3 + 0.755x_9^2 \\
V_2 \;=\;& 0.546x_{15}^2 + 0.0114x_{15}x_{16} + 0.789x_{16}^2 + 0.388x_{15}x_6 - 0.0151x_{16}x_6 + x_6^2 \\
&-0.101x_{15}x_{10} + 0.0372x_{16}x_{10} + 0.228x_6x_{10} + 0.611x_{10}^2 + 0.00175x_{15}x_2 \\
&+0.0215x_{16}x_2 + 0.0138x_6x_2 - 0.00344x_{10}x_2 + 0.795x_2^2
\end{aligned}$$

$$\begin{aligned}
V_3 \;=\;& 0.719x_4^2 + 0.500x_4x_5 + 0.338x_5^2 + 0.00238x_{14}x_4 - 0.00443x_{14}x_5 + 0.533x_{14}^2 \\
&+0.0653x_{12}x_4 + 0.0126x_{12}x_5 + 0.438x_{14}x_{12} + 0.755x_{12}^2 - 0.139x_{13}x_4 \\
&+0.0163x_5x_{13} + 0.0255x_{14}x_{13} + 0.0305x_{13}x_{12} + 0.784x_{13}^2 + 0.0220x_{11}x_4 + \\
&0.000297x_{11}x_5 - 0.000639x_{14}x_{11} + 0.00761x_{12}x_{11} - 0.000114x_{13}x_{11} + 0.0806x_{11}^2
\end{aligned}$$

The multipliers that verify $V_c(x) = V_1 + \alpha_1 V_2 + \alpha_2 V3$ is a CoLF are $\alpha_1 = 0.226e{-}4$ and $\alpha_2 = 0.542$. The full set of system equations are given in the Appendix at the end of this chapter.

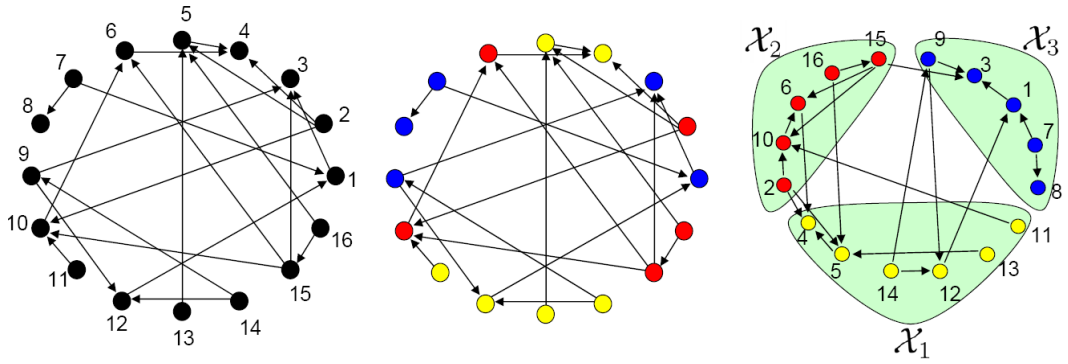


Figure 4.2: Lotka-Voltera system interaction structure (left) decomposed into 3 subsystems (middle) and reordered (right).

Also shown in Table 4.1 is the cost of computing sparse Lyapunov functions denoted $\widehat{V}_i$ using the methodology described in Section 4.3. As expected the subsystem Lyapunov function construction takes slightly longer, however, when verifying the composite function (which is where the bulk of the processing occurs) we achieve a reduction of over 50% in computational time.

Non-sparse	cpu time (s)	Sparse	cpu time (s)
V_1	0.25	$\widehat{V}_1$	0.38
V_2	0.25	$\widehat{V}_2$	0.37
V_3	0.44	$\widehat{V}_3$	0.59
V_c	1415.23	$\widehat{V}_c$	688.54

Table 4.1: Computational cost of constructing a Lyapunov function.

4.4.3 Overlapping Decomposition

Implicit in the CoLF framework is the fact that each of the subsystems must be stable, i.e. a Lyapunov function for the subsystem exists. In some instances the decomposition algorithm may produce an unstable subsystem. This is likely to be the case for systems with strong coupling between all states. In such instances composite methods will fail.

Assume that a decomposition of (4.1) into (4.2) has been performed using for example the methods presented in Section 4.4.2. For simplicity, let us consider the case of a system bisection where we have two subsystems $S(\mathcal{T}_1)$ and $S(\mathcal{T}_2)$

where $S(\mathcal{T}_i)$ corresponds to $\dot{x}_i = f_i(x_i)$. Further assume that $S(\mathcal{T}_1)$ is unstable and $S(\mathcal{T}_2)$ is stable.[1] Under these conditions composite stability methods cannot be applied. However, if we drop the requirement that $\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset$ and allow subsystem $S(\mathcal{T}_1)$ to include states that also belong to the stable subsystem $S(\mathcal{T}_2)$ then the new subsystem which we denote $S(\widehat{\mathcal{T}}_1)$ may well be stable [105, 106]. Intuitively this may occur when a state that has been decomposed and placed into one subsystem is also strongly connected to states in another subsystem.

We propose a simple iterative procedure reminiscent of the KL algorithm [107] that appends states from $\mathcal{T}_1$ to $\mathcal{T}_2$ at each iteration testing the stability of the new system $S(\widehat{\mathcal{T}}_1)$, removing the appended state if $S(\widehat{\mathcal{T}}_1)$ is still unstable. The algorithm is presented in Table 4.2.

Decompose $\mathcal{T}$ *into* $\mathcal{T}_1$, $\mathcal{T}_2$
If $S(\mathcal{T}_1)$ *is unstable*
  $\widehat{\mathcal{T}}_1 \leftarrow \mathcal{T}_1$
  For $i = 1$ *to card*$(\mathcal{T}_2)$
    $\widehat{\mathcal{T}}_1 \leftarrow \widehat{\mathcal{T}}_1 \bigcup \mathcal{T}_2^i$
    If $S(\widehat{\mathcal{T}}_1)$ *is stable*
      Find composite Lyapunov function for $S(\widehat{\mathcal{T}}_1)$, $S(\mathcal{T}_2)$
      Break
    Else $\widehat{\mathcal{T}}_1 \leftarrow \mathcal{T}_1$; *End If*
  End For
End If

Table 4.2: The Overlapping Decomposition Algorithm

Remark 4 *It is only necessary to iterate through the states in* $\mathcal{T}_2$ *that are neighbours of those in* $\mathcal{T}_1$.

Clearly it is also possible to allow for multiple states to be appended to the unstable subsystem in the hope of stabilizing it. In terms of implementation such an approach is trivial, however by following this path the combinatorial nature of the problem should be obvious.

[1] Of course we are referring to the *equilibrium point* at the origin being stable or unstable.

Consider the 12-state nonlinear dynamical system given in the Appendix to this chapter. The system has an equilibrium point at the origin which is asymptotically stable by Lyapunov's indirect method. Due to the high state dimension, SOS programming cannot be used directly to verify stability of the nonlinear system and there is no obvious structure that we can exploit. Using the decomposition algorithm presented in Section 4.4.2 the system is decomposed into two subsystems with state partitions:

$$\mathcal{T}_1 = \{x_3, x_4, x_5, x_6, x_8, x_{10}\}, \quad \mathcal{T}_2 = \{x_1, x_2, x_7, x_9, x_{11}, x_{12}\}.$$

The zero equilibrium of $S(\mathcal{T}_2)$ is locally asymptotically stable as certified by a quadratic Lyapunov function obtained using SOSTOOLS. Stability of the equilibrium of $S(\mathcal{T}_1)$ could not be verified, thus preventing a composite analysis. By allowing a non-disjoint state partition, specifically letting $\text{card}(\mathcal{T}_1 \cap \mathcal{T}_2) = 1$ where the common state originates from $\mathcal{T}_2$ it is possible to find a Lyapunov function for the new subsystem $S(\widehat{\mathcal{T}}_1)$. Applying the iterative algorithm described in Table 4.2 we find that appending any state from the set $\mathcal{Z} = \{x_1, x_2, x_{11}, x_{12}\}$ to $\mathcal{T}_1$ makes $S(\widehat{\mathcal{T}}_1)$ stable. In each instance a quadratic subsystem Lyapunov function was found. Proceeding with a composite stability analysis a CoLF can be constructed when the appended state to $\mathcal{T}_1$ is a member of the subset of $\mathcal{Z}$ given by $\widehat{\mathcal{T}} = \{x_1, x_{11}, x_{12}\}$. The CoLF function consists of the two subsystem Lyapunov functions given below for $\widehat{\mathcal{T}}_1 = \mathcal{T}_1 \cup \{x_1\}$:

$$\begin{aligned} V_1 \;=\; & 1.59x_8x_4 - 1.52x_5x_4 + 1.35x_3^2 + 2.03x_{10}^2 + 2.33x_8^2 + 2.42x_5^2 + 2.17x_6^2 \\ & +3.77x_4^2 + 2x_1^2 - 0.899x_5x_8 + 0.993x_4x_3 - 0.181x_4x_6 - 1.08x_4x_{10} \\ & +1.04x_8x_1 + 0.182x_8x_3 + 0.627x_8x_6 - 0.250x_8x_{10} - 0.641x_5x_3 + 0.793x_5x_6 \\ & +0.621x_5x_{10} - 0.321x_{10}x_3 - 0.432x_{10}x_6 + 0.652x_1x_3 + 0.989x_1x_4 \\ & -0.755x_1x_{10} - 0.286x_1x_5 - 0.334x_1x_6 - 0.524x_6x_3 \end{aligned}$$

$$\begin{aligned} V_2 &= 1.86x_9^2 + 0.355x_9x_2 + 2.09x_2^2 + 0.0663x_1x_9 - 0.226x_1x_2 + 1.49x_1^2 \\ &\quad -0.04751x_9x_{11} + 0.101x2x_{11} + 0.383x_1x_{11} + 2.16x_{11}^2 + 0.602x_9x_7 \\ &\quad -0.363x_2x_7 + 0.0977x_1x_7 + 0.216x_{11}x_7 + 1.50x_7^2 + 0.143x_9x_{12} \\ &\quad -0.0914x_{12}x_2 - 0.261x_1x_{12} - 0.388x_{12}x_{11} + 0.0748x_{12}x_7 + 2.23x_{12}^2. \end{aligned}$$

We take this opportunity to demonstrate the sparse Lyapunov function algorithm described in Section 4.3. Applying the algorithm to subsystem $S(\mathcal{T}_2)$ produces a sparse subsystem Lyapunov function $\widehat{V}_2$:

$$\begin{aligned} \widehat{V}_2 &= 0.101x_9^2 + 0.1x_2^2 + 0.1x_1^2 - 0.000445x_{11}x_9 + 0.1x_{11}^2 + 0.12x_7^2 \\ &\quad +0.00277x_{12}x_9 - 0.00121x_{12}x_{11} + 0.102x_{12}^2. \end{aligned}$$

Computing the subsystem Lyapunov functions took approximately 35s per subsystem on a standard desktop computer. To verify the derivative condition of the CoLF took less than 50s. To put this in perspective, the SDP solver was unable to execute a single iteration before running out of memory using a direct SOS approach.

4.5 Remarks on Tractability

Before concluding this chapter we highlight the computational intractability of a direct solution to the system decomposition problem in the context of the composite Lyapunov function framework.

Essentially the problem of finding a system decomposition such that subsystem Lyapunov functions verify the stability of the original system hinges on being able to solve two non-convex optimization problems simultaneously:

1. Solution of a Bilinear Matrix Inequality (BMI)

2. Optimization over the space of permutation matrices

We now describe in detail why the first problem makes system decomposition both theoretically and numerically challenging.

The second point has been addressed from a geometric point of view using Birkhoff polytopes, see [108] and the references therein. Such polytopes have recently been studied from a computational point of view using semidefinite programming relaxations [109]. Informally, a Berkhoff polytope is the convex hull of the set of permutation matrices of fixed dimension. This issue is discussed in more detail in Section 6.6 where optimization over Π appears in the form of a *clustering* based decomposition algorithm.

Bilinear Matrix Inequalities

Many problems related to controller synthesis can be formulated as BMI optimizations, unfortunately these result in non-convex problems [110].

The optimization problem we would ideally like to be able to solve is, given a Hurwitz matrix $\widehat{F} \in \mathcal{R}^{n\times n}$ find matrices $T \in \Pi_n$ and $P \in \mathcal{S}^n_{++}$ such that

$$F^T P + PF \prec 0 \tag{4.30a}$$

$$P = \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix} \succ 0 \tag{4.30b}$$

$$F = T\widehat{F}T^{-1} \tag{4.30c}$$

is feasible.

Substituting (4.30c) into (4.30a) and using $T^{-1} = T^T$ gives

$$TA^T T^T P + PTAT^T \prec 0$$

which is not convex in P and T. Imposing the block structure from (4.30b) on

the state permutation gives

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{12} & F_{22} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \widehat{F}_{11} & \widehat{F}_{12} \\ \widehat{F}_{21} & \widehat{F}_{22} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}^T. \tag{4.31}$$

Expanding the F_{11} block we obtain

$$F_{11} = T_{11}\widehat{F}_{11}T_{11}^T + T_{11}\widehat{F}_{12}T_{12}^T + T_{12}\widehat{F}_{21}T_{11}^T + T_{12}\widehat{F}_{22}T_{12}^T. \tag{4.32}$$

At this stage the sub-matrices T_{ij} are no longer permutation matrices. In fact all we can say about the T_{ij} matrices is that they are binary and due to the orthogonality properties of T the following non-convex relationships must hold:

$$TT^T = I \Leftrightarrow \begin{cases} T_{11}T_{11}^T + T_{12}T_{12}^T = I \\ T_{21}T_{21}^T + T_{22}T_{22}^T = I \\ T_{21}T_{11}^T + T_{22}T_{12}^T = 0 \\ T_{11}T_{21}^T + T_{12}T_{22}^T = 0 \end{cases}, \quad T^TT = I \Leftrightarrow \begin{cases} T_{11}^TT_{11} + T_{21}^TT_{21} = I \\ T_{12}^TT_{12} + T_{22}^TT_{22} = I \\ T_{12}^TT_{11} + T_{22}^TT_{21} = 0 \\ T_{11}^TT_{12} + T_{21}^TT_{22} = 0 \end{cases}.$$

Combining the orthogonality constraints above with the block structure in (4.32) and the block Lyapunov inequalities (4.30a–4.30b) it very quickly becomes apparent that solving (4.30) directly is not possible and deriving a convex relaxation is nontrivial as we will now illustrate.

4.6 Conclusion

In this chapter the concept of dynamical system decomposition has been presented. The stability analysis that followed was based on the composite Lyapunov function framework. A method for reducing the computational complexity of computing a sum of squares decomposition for testing the CoLF derivative inequality was then described.

It was shown how a system decomposition that minimizes the *energy* flow

between subsystems can facilitate a CoLF based analysis. The decomposition framework we presented allowed for the case of both disjoint and non-disjoint subsystems. Further computational savings were achieved by including the sparse Lyapunov function algorithm for the case of nonlinear systems. Throughout the chapter, the algorithms we derived were illustrated on a variety of systems including an Internet congestion control scheme, an ecological network and numerous numerical examples.

4.7 Appendix

Lotka-Volterra system dynamics from Section 4.4.2:

$$
\begin{aligned}
\dot{x}_1 &= -x_1^2 - 0.167x_1x_3 - 1.16x_1 - 0.193x_3 \\
\dot{x}_2 &= -x_2^2 - 0.969x_4x_2 - 0.264x_5x_2 - 0.113x_{10}x_2 - 1.26x_2 - 1.22x_4 - 0.333x_5 \\
&\quad -0.142x_{10} \\
\dot{x}_3 &= -x_3^2 - 0.0397x_3 \\
\dot{x}_4 &= -x_4^2 - 0.0926x_4 \\
\dot{x}_5 &= -x_5^2 - 0.847x_4x_5 - 1.77x_4 - 2.09x_5 \\
\dot{x}_6 &= -0.866x_6x_4 - x_6^2 - 0.271x_4 - 0.313_x6 \\
\dot{x}_7 &= -0.143x_1x_7 - x_7^2 - 0.388x_8x_7 - 0.163x_1 - 1.14x_7 - 0.443x_8 \\
\dot{x}_8 &= -x_8^2 - 0.198x_8 \\
\dot{x}_9 &= -0.146x_9x_3 - x_9^2 - 0.0136x_9x_{12} - 0.107x_3 - 0.730x_9 - 0.00994x_{12} \\
\dot{x}_{10} &= -0.419x_6x_{10} - x_{10}^2 - 0.822x_6 - 1.96x_{10} \\
\dot{x}_{11} &= -0.0947x_{11}x_{10} - x_{11}^2 - 0.00370x_{10} - 0.0391x_{11} \\
\dot{x}_{12} &= -0.801x_1x_{12} - x_{12}^2 - 0.195x_1 - 0.244x_{12} \\
\dot{x}_{13} &= -0.211x_5x_{13} - x_{13}^2 - 0.193x_5 - 0.915x_{13} \\
\dot{x}_{14} &= -0.571x_{14}x_9 - 0.709x_{14}x_{12} - x_{14}^2 - 0.881x_9 - 1.09x_{12} - 1.54x_{14} \\
\dot{x}_{15} &= -0.548x_{15}x_3 - 0.796x_{15}x_6 - 0.230x_{15}x_{10} - x_{15}^2 - 0.936x_3 - 1.36x_6 - 0.394x_{10} \\
&\quad -1.71x_{15} \\
\dot{x}_{16} &= -0.407x_{16}x_5 - 0.527x_{15}x_{16} - x_{16}^2 - 0.275x_5 - 0.356x_{15} - 0.676x_{16}
\end{aligned}
$$

Dynamical system considered in Section 4.4.3:

$$
\begin{aligned}
\dot{x}_1 &= -1.57x_1 - 0.574x_2 - 0.583x_3 + 0.942x_4 + 1.35x_6 + 0.861x_7 + 1.51x_9 + 1.10x_{11} \\
&\quad +1.14x_{12} + x_2x_6x_9 \\
\dot{x}_2 &= -0.760x_2 + 1.54x_3 - 1.63x_4 - 0.819x_5 - 0.925x_6 - 1.06x_7 - 0.535x_{10} \\
&\quad +0.681x_{11} + x_7x_{10}x_{11} \\
\dot{x}_3 &= 1.77x_1 - 0.983x_2 - 0.766x_3 - 0.930x_6 - 1.12x_9 - 0.998x_{10} + 1.84x_{11} \\
&\quad +0.772x_{12} + x_3^3 \\
\dot{x}_4 &= -1.37x_1 + 1.31x_2 + 0.790x_3 - 0.572x_4 - 1.07x_5?0.783x_6 - 0.938x_8 + 2.03x_9 \\
&\quad -0.857x_{11} + x_4x_8x_{11} \\
\dot{x}_5 &= -1.04x_1 + 0.755x_2 + 1.12x_4 - 0.538x_5 - 0.563x_8 + 1.40x_9 - 1.32x_{11} + x_1x_2x_5 \\
\dot{x}_6 &= 1.26x_2 + 1.11x_4 - 0.885x_6 - 0.935x_8 + 1.05x_9 - 1.30x_{11} + x_2x_4x_{11} \\
\dot{x}_7 &= -0.845x_1 - 1.67x_7 + 1.72x_9 + 0.839x_{10} - 1.25x_{11} - 0.600x_{12} + x_1x_7x_{11} \\
\dot{x}_8 &= 0.583x_1 + 1.20x_4 + 0.816x_5 + 0.599x_6 - 0.735x_9 + x_1^3 \\
\dot{x}_9 &= -1.25x_1 + 1.04x_2 + 0.990x_3 - 1.44x_4 - 1.44x_5 - 1.29x_6 - 1.20x_9 - 1.16x_{10} \\
&\quad -3.15x_{11} - 0.998x_{12} + x_1x_6x_{10} \\
\dot{x}_{10} &= 1.43x_2 + 0.835x_3 + 0.545x_7 - 0.890x_8 - 0.973x_{10} + x_2x_3x_8 \\
\dot{x}_{11} &= -2.29x_3 + 1.41x_4 + 1.43x_5 + 0.512x_6 + 2.00x_7 - 0.801x_8 + 2.46x_9 - 0.659x_{10} \\
&\quad -0.833x_{11} + 1.29x_{12} + x_6^3 \\
\dot{x}_{12} &= -0.850x_1 - 0.919x_3 + 0.515x_7 + 1.20x_9 - 1.25x_{11} + x_1x_9x_{11}
\end{aligned}
$$

Chapter 5

Decomposition of the EGF-MAPK Signaling Pathway

In this chapter the decomposition algorithm described in Chapter 4 is applied to a model of the Epidermal Growth Factor receptor induced MAP kinase (EGF-MAPK) signaling pathway. The system model considered is typical of the type of high-fidelity models that arise in systems biology. The dimensionality coupled with the nonlinearity make it difficult to analyse using conventional techniques. A simplified implementation of the model consists of 35 coupled differential equations with over 70 parameters, some of which are uncertain, and includes states that evolve over multiple time scales.

5.1 Pathway Model

The model we consider describes the complete signaling cascade of the EGF receptor-induced MAP kinase pathway [111]. The EGFR-MAPK pathway is probably the most studied signal transduction pathway in mammalian cells and is responsible for cell proliferation, differentiation and migration. Epidermal Growth Factor receptors are known to play an important role during embryonic and postnatal development as well as in tumor progression.

Work in [112] identifies the following key processes taking place in the pathway:

1. The EGFs bind to the Epidermal Growth Factor Receptors (EGFR) and activate them with every two receptors becoming a dimer.

2. The cofactors GAP, Shc (optional), Grb2 and Sos bind to the dimer to form a complex.

3. The complex then activates nearby Ras-GTP on the cell membrane. Raf binds to the activated RAS-GTP and then drives phosphorylation of MEK followed by ERK.

4. Finally, phosphorylated ERK goes into the nucleus and drives the downstream gene expression.

The details of this pathway and the original model for this system were presented in [111], one graphical depiction of the pathway is shown in Figure 5.1. This was a 94-state (species) system of nonlinear polynomial differential equations containing over 100 kinetic parameters. Work reported in [112, 113] made an assumption to ignore receptor internalization in the original model. The model without receptor internalization was used as the starting model for reduction in [112, 113]. This reduced model has 35-states, is nonlinear and is the model we will apply our decomposition algorithm to. The reader is referred to the previous papers for the full ODE model.

5.2 Results

The goal of system decomposition in this case is twofold. In the first instance we wish to uncover the modularity present in the system from the differential equations. This decomposition will take into account the dynamics of the system (by minimizing the energy flow between subsystems) rather than just a purely

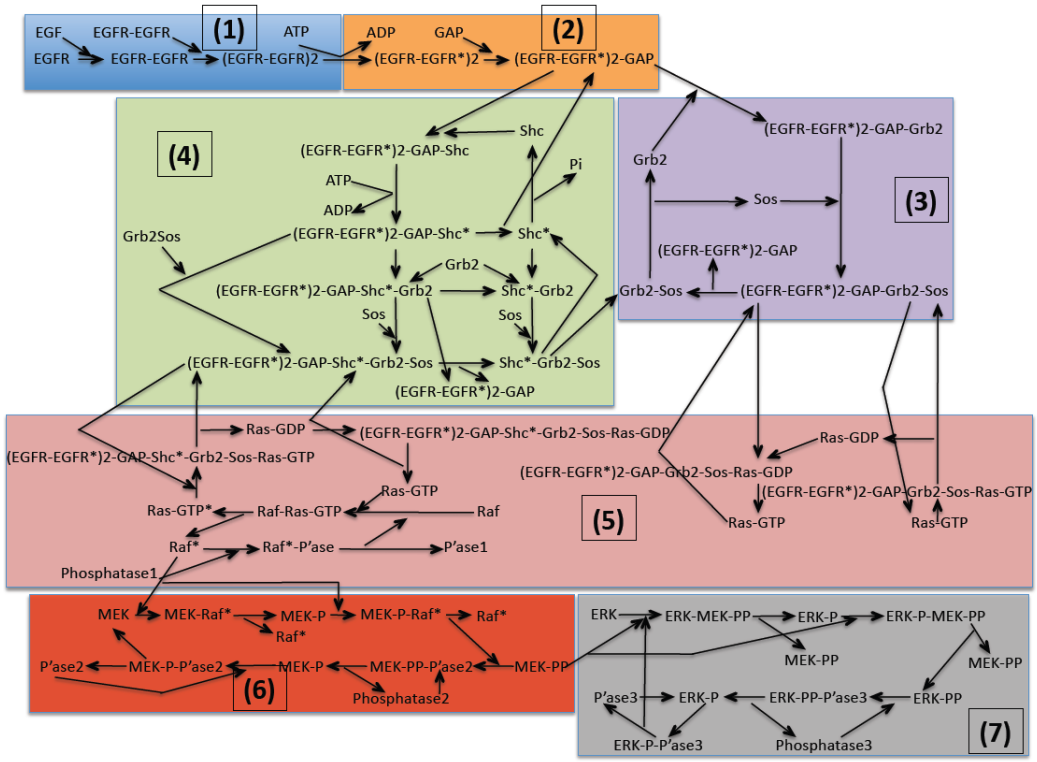


Figure 5.1: Constituent parts of the EGF-MAPK pathway. 1: EGF binding, 2: Signaling complex formation without Shc (3), and with Shc (4), 5, 6: MAPK cascade broken down into two stages corresponding to the formation of an MEK complex (5) and an ERK complex (6).

topological decomposition based on a static graph. Ideally the decomposition will uncover subsystems that have some biological significance although there is no guarantee of this. The second goal of the decomposition is to verify the stability of the 35-state nonlinear model through a composite Lyapunov function.

The presence of stability in biological models is important for several reasons. In a real biological system unstable dynamics imply that the concentration of a certain complex can increase at an arbitrarily fast rate and to a potentially infinite level. Obviously this is not physically possible as unmodeled dynamics will give rise to saturation effects. Thus if the model is unstable and the real system is clearly stable then instability can be used to help refine the model. Alternatively it could be the case that the real system is in competition with another system and instability leads to extinction. The type of stability considered is also important,

many biological systems have oscillatory dynamics in which case they will not be asymptotically stable but may be stable in a Lyapunov sense.

As a benchmark we compare our results to those obtained in [113]. These benchmark results were obtained after a manual analysis based on model reduction, time-scale separation and minimization of retroactivity. In contrast our decomposition algorithm is automated and was recursively called until the pathway was broken down into 7 subsystems as shown in Figure 5.2.

The resulting decomposition is in strong agreement with the modularization found using retroactivity (up and down-stream feedback) based methods [114] and analytic methods [113]. In particular the MAPK cascade is identified and the MEK and ERK blocks are identified and isolated (blocks 5 and 6 in Figure 5.2). The Raf and Ras activation cycles are identified (blocks 4 and 5 in Figure 5.2) and the EGF binding process is found (block 1 in Figure 5.2).

The manual decomposition in [113] identified 7 subsystems: EGF binding, signaling complex formation (both Shc independant and Shc dependent pathways), Raf activation, Ras activation and the MAPK cascade which contains the MAP kinase kinase module (ERK) and the MAP kinase module (MEK). The only notable difference between the decomposition produced by our automated algorithm and that of [113] is that our method lumps EGF binding and the signaling complex formation together (blocks 1 and 2 in Figure 5.1) and breaks the Raf and Ras activation into three separate modules. In contrast, the manual approach keeps the Raf and Ras activation process as one module. We observe however that this module actually consists of two unconnected components, thus from a computational point of view our decomposition is likely to be more beneficial.

Furthermore, each of the subsystems the decomposition algorithm uncovers is stable as verified by a polynomial Lyapunov function constructed using sum of squares programming. For the linearized version of the system, we were able to construct a composite Lyapunov function that verifies the stability of the original large-scale system. For the nonlinear case, using sparse subsystem Lyapunov

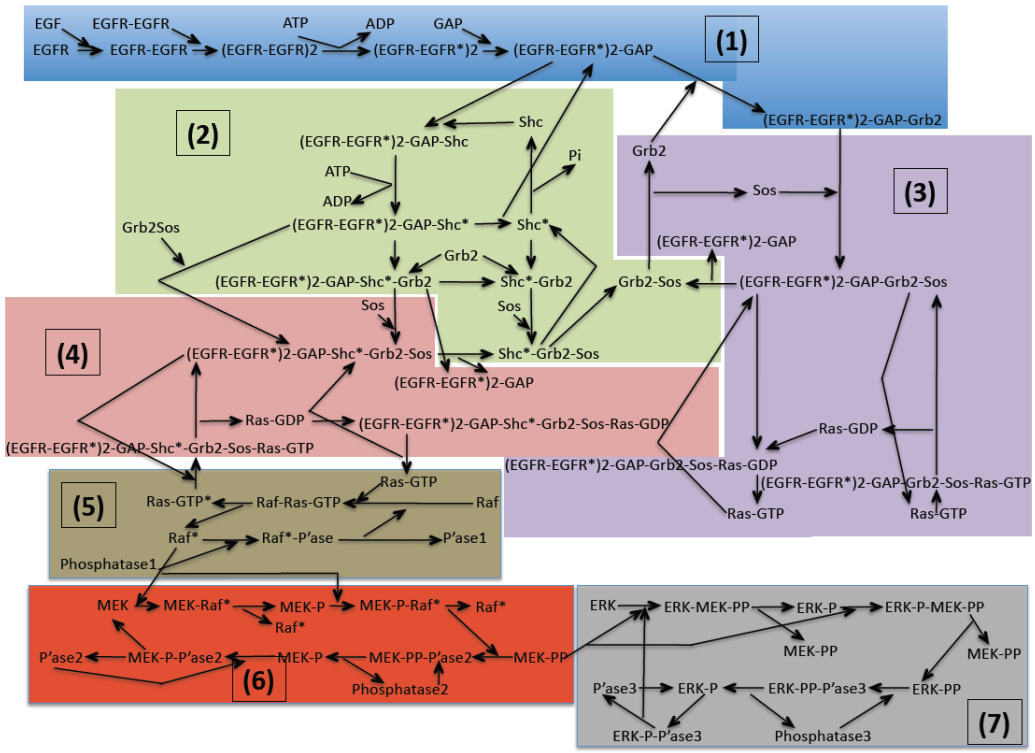


Figure 5.2: Applying the decomposition algorithm presented in the previous chapter we identify the subsystems shown above.

functions it was possible to construct a composite Lyapunov function, although there was a significant level of numerical error.[1] However, for six of the seven subsystems a composite Lyapunov function with normal levels of numerical error was constructed.

5.3 System Decomposition With Uncertainty

We now turn our attention to applying the decomposition algorithm to dynamical systems that contain uncertainty. Such models frequently occur in mathematical biology in the context of modeling biochemical reaction networks where the kinetic constants are uncertain. In particular the systems we consider take the

[1] In this instance the error referred to is the residual error computed by the SDP solver SeDuMi.

form

$$\dot{\hat{x}} = \hat{f}(\hat{x}, p) \tag{5.1}$$

where $\hat{x} \in \mathcal{R}^n$ is the state vector and $p \in \mathcal{P}$ parameterizes the model where

$$\mathcal{P} = \left\{ p \in \mathcal{R}^{n_p} \mid \underline{p_i} \leq p_i \leq \overline{p_i},\ i = 1, \ldots, n_p \right\},$$

and the initial condition is $\hat{x}(0) = \hat{x}_0$. We came across this type of system in Chapter 3. Performing stability analysis of (5.1) directly is still a challenging task even using sum of squares methods. A more computationally attractive approach is to develop a Linear Parameter Varying (LPV) representation of (5.1). LPV models take the form

$$\dot{\hat{x}} = \widehat{F}(\theta)\hat{x} \tag{5.2}$$

where the state matrix $\widehat{F}(\theta)$ depends affinely on $\theta \in \Theta$ where

$$\widehat{F}(\theta) = \widehat{F}_0 + \sum_{i=1}^{k} \theta_i \widehat{F}_i$$

with $\widehat{F}_i$ known and fixed and

$$\Theta = \left\{ \theta \in \mathcal{R}^k \mid \underline{\theta_i} \leq \theta_i \leq \overline{\theta_i},\ i = 1, \ldots, k \right\}.$$

The LPV model (5.2) of the EGF-MAPK pathway from the nonlinear system (5.1) was developed using the methodology described in [115], we omit the details here. The uncertain parameters in the model are determined by performing a sensitivity analysis and selecting the three most sensitive parameters in the model. We allow these parameters to vary by $\pm 5\%$ of their nominal value. It turns out that larger deviations make the system unstable, this is a property of the model not the algorithm.

The objective of robust system decomposition is to convert (5.2) into two

subsystems of smaller state dimension connected in feedback in such a manner that allows us to conclude stability of the original system from the decomposed, isolated subsystems. The subsystems are given by

$$\begin{aligned} \dot{x}_1 &= F_{11}(\theta_1)x_1 + F_{12}(\theta_1)u_1, \quad y_1 = x_1 \\ \dot{x}_2 &= F_{21}(\theta_2)x_1 + F_{22}(\theta_2)u_2, \quad y_2 = x_2 \\ \theta_1 &\in \Theta_1, \quad \theta_2 \in \Theta_2, \quad \Theta_1, \Theta_2 \subseteq \Theta \end{aligned} \tag{5.3}$$

where $y_1 = u_2$ and $y_2 = u_1$ and the matrices $F_{ij}(\theta_k)$ are affine in θ_k. Previously the objective was to partition the states to minimize the energy flow between subsystems. In the LPV case it is also desirable to partition the system parameters i.e. the resulting subsystems should share as few parameters as possible. This is an attractive option as it leads to simpler models for the subsystems which in turn is helpful if a robust stability analysis is to be carried out [116].

The notable difference between parameter and state partitioning is that the parameter partitions do not necessarily have to be disjoint, in fact such a partition is likely to be impossible. It would however be desirable to minimize the number of parameters in the intersection of the two sets if a computational stability analysis is to be performed.

The first stage of the robust decomposition algorithm proceeds in exactly the same manner as described in Section 4.4.2 which we summarize here for convenience:

1. Construct a graph $\mathcal{G}_0(\mathcal{V}, \mathcal{E}_0)$ corresponding to the nominal part of system (5.2) and append the output map $y = \mathcal{C}^T(\mathcal{G}_0)\hat{x}$:

$$\begin{aligned} \dot{\hat{x}} &= \widehat{F}_0\hat{x}, \\ y &= \mathcal{C}^T(\mathcal{G}_0)\hat{x}. \end{aligned} \tag{5.4}$$

2. Construct the energy matrix $\mathcal{W}^*(\mathcal{G}_0)$ from (5.4) and hence the Laplacian

matrix $\mathcal{L}(\mathcal{G}_0)$.

At this point it is possible that the magnitudes of the entries of the two Laplacians may be of differing orders of magnitude and thus the partitioning algorithm is not likely to return a useful result as the problem is ill-conditioned. For this reason it may be beneficial to normalize $\mathcal{L}(\mathcal{G}_o)$. Let $\mathcal{D}(\mathcal{G}_0) \triangleq \text{diag}(\mathcal{W}^*(\mathcal{G}_0)\mathbf{1})$, then the normalized Laplacian [35] is given by

$$\left[\widetilde{\mathcal{L}}(\mathcal{G}_0)\right]_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } [\mathcal{D}(\mathcal{G}_0)]_{ii} \neq 0 \\ -\frac{1}{\sqrt{[\mathcal{D}(\mathcal{G}_0)]_{ii}[\mathcal{D}(\mathcal{G}_0)]_{jj}}} & \text{if } (v_i, v_j) \in \mathcal{E}_0 \\ 0 & \text{otherwise} \end{cases}.$$

Alternatively, $\widetilde{\mathcal{L}}(\mathcal{G}_0)$ can be computed directly from $\mathcal{L}(\mathcal{G}_0)$:

$$\widetilde{\mathcal{L}}(\mathcal{G}_0) = \mathcal{D}^{-1/2}(\mathcal{G}_0)L(\mathcal{G}_0)\mathcal{D}^{-1/2}(\mathcal{G}_0).$$

One of the useful properties of normalized Laplacians is that their spectrum is such that all the eigenvalues of $\widetilde{\mathcal{L}}(\mathcal{G}_0)$ lie in $[0, 2]$.

With the normalized Laplacian matrix constructed we can proceed with the robust decomposition algorithm. Typically the matrices $\widehat{F}_i$ are sparse (at least in comparison to $\widehat{F}_0$) indicating that not all parameters affect every element in the nominal system matrix. The motivation is to include parameter partitioning with the state partitioning algorithm. To do so a method for combining the information from the matrices $\widehat{F}_i, i = 1, \ldots, k$ needs to be incorporated into the graph $\mathcal{G}_0$.

For each matrix $\widehat{F}_i$, $i = 1, \ldots, k$ a graph $\mathcal{G}_i$ with associated binary adjacency matrices $A_i(\mathcal{G}_i)$ is created in the following manner:

1. Represent each state as a vertex.

2. Construct a binary adjacency matrix (not necessarily symmetric) such that $[\mathcal{A}_i(\mathcal{G}_i)]_{jk} = 1$ if $[\widehat{F}_i]_{kj} \neq 0$.

With the *parameter graphs* constructed let $\mathcal{A}(\mathcal{G}_\theta) = \frac{1}{k}\sum \mathcal{A}_i(\mathcal{G}_i)$ and form the (normalized) parameter Laplacian matrix denoted by $\widetilde{\mathcal{L}}(\mathcal{G}_\theta)$. The total state-parameter dependence for (5.2) is then captured by the graph $\mathcal{G}^*$ which corresponds to the Laplacian matrix obtained by forming the weighted sum of the state and parameter graphs:

$$\mathcal{L}(\mathcal{G}^*) = \widetilde{\mathcal{L}}(\mathcal{G}_0) + \gamma\widetilde{\mathcal{L}}(\mathcal{G}_\theta)$$

where $\gamma > 0$ is a tuning parameter.

The parameter γ is used to weight the uncertainty dependence versus energy flow in the decomposition, in essence make sure they are of the same magnitude so that the partitioning algorithm is acting on a relevant graph. By setting $\gamma = 0$ the decomposition algorithm from Section 4.4.2 is recovered. Once $\mathcal{L}(\mathcal{G}^*)$ has been constructed the algorithm proceeds in the usual manner. Setting $\gamma = 1$ covers the nominal case where there is no preference given to parameter uncertainty, this is the suggested starting point for tuning.

The intended use of the uncertain decomposition algorithm is to uncover structural relationships between parameter and state dependence in a given system. By adjusting γ the decomposition places more or less emphasis on parameter dependence versus energy flow. In principle the algorithm could then be used for stability analysis by incorporating the results developed by Gahinet *et al* [116] on affine parameter dependent Lyapunov function and parametric uncertainty, although we do not pursue this here.

Applying this uncertain decomposition algorithm to the EGF-MAPK model from the previous section we obtain a decomposition that is consistent with the nominal decomposition for the case of $\gamma = 1$. Additionally, the same decomposition was achieved for values of $\gamma \in [0.8, 1.95]$ implying that the parametric uncertainty doesn't affect the decomposition of this system much.

5.4 Conclusion

In this chapter we have shown that the decomposition algorithm can be used to uncover structure as well as verify stability using composite Lyapunov functions. The class of systems that the decomposition can be applied to for uncovering structure is then extended to include uncertain systems that can be modeled in LPV framework. These results are illustrated on a 35-state model of the EGF-MAPK signaling pathway.

Chapter 6

System Decomposition via Dissipation Inequalities

We now take an alternative approach to stability verification via system decomposition. The methods presented in this chapter are related to those presented in Chapter 4 except that here we use storage functions and supply rates to explicitly take into account the interaction between the decomposed subsystems. An advantage of this type of approach is that the resulting stability tests are typically less computationally demanding than the composite Lyapunov function methods described previously. In particular we will look to use *small gain* and *passivity* based stability proofs which provide a more natural framework for assessing the stability of interconnected systems.

We begin by introducing the concept of dissipation inequalities and storage functions which extend the concept of Lyapunov functions for the case of systems that contain inputs and outputs. The methods are introduced by demonstrating how they can be used in conjunction with a decomposition algorithm to obtain bounds on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of a system. The rest of the chapter then focusses on deriving stability tests for interconnected systems via dissipation inequalities. The chapter is concluded with the presentation of a clustering algorithm for obtaining system decompositions with desirable stability/performance properties.

6.1 Problem Formulation

Once the concept of dissipation inequalities is introduced in the following section the focus of this chapter will be on how the stability of the system $\dot{x} = f(x)$ can be analyzed once it has been decomposed by explicitly taking into account the input and output dynamics. This approach is in contrast to work in the previous chapter where the emphasis was placed on obtaining a system decomposition that would allow us to essentially ignore the coupling between decomposed subsystems.

By way of introduction we will first address how to compute the $\mathcal{L}_2 \to \mathcal{L}_2$ performance of the system

$$\begin{aligned} \dot{x} &= f(x) + G(x)u, \quad x(0) = x_0 \\ y &= h(x) \end{aligned} \tag{6.1}$$

via dissipation inequalities corresponding to a decomposition of (6.1) into M subsystems. Of course the description of the subsystems will be more complicated than (4.2) as there will be coupling in the state, input and output.

We will then investigate various stability metrics based on passivity when the LTI system

$$G \begin{cases} \dot{\hat{x}} = \widehat{A}\hat{x}, \quad \hat{x}(0) = \hat{x}_0 \\ \hat{y} = \hat{x} \end{cases} \tag{6.2}$$

is decomposed into $M = 2$ subsystems (of conformable dimension) connected in feedback such that

$$G_1 \begin{cases} \dot{x}_1 = A_{11}x_1 + u_1 \\ u_1 = A_{12}x_2 \\ y_1 = x_1 \end{cases}, \quad G_2 \begin{cases} \dot{x}_2 = A_{22}x_2 + u_2 \\ u_2 = A_{21}x_1 \\ y_2 = x_2 \end{cases}, \tag{6.3}$$

where $x_i \in \mathcal{R}^{n_i}$, $x = T\hat{x}$ with $T \in \Pi_n$ and no state belongs to multiple subsystems. The interconnection between the subsystems G_1 and G_2 is described by

the interconnection operator

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}}_{H} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{6.4}$$

The framework easily generalizes to multiple decompositions ($M > 2$) but for clarity we will focus on the 2 subsystem case.

6.2 Preliminaries

In this section two important concepts will be introduced. Dissipation inequalities and the small gain theorem. Dissipation inequalities combined with storage functions and supply rate functions are used to generalize the notion of Lyapunov stability to the case of systems that contain inputs and outputs. The small gain theorem, a cornerstone in robust control theory, is used to verify stability of systems connected in feedback, we will now describe each of these in turn..

6.2.1 Dissipation Inequalities

Consider the Linear Time Invariant system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{6.5}$$

Dropping the time dependence from our notation, we have $x \in \mathcal{R}^n$, $u \in \mathcal{R}^p$ and $y \in \mathcal{R}^m$ denoting the state, input and output respectively. Observe that the subsystems G_i in (6.3) are realizations of (6.5) with $C = I$. We are interested in characterizing the *dissipativity* [75] of system (6.5) via quadratic supply functions and (Q, S, R) supply rates using the methods of Moylan and Hill [74].

System (6.5) is said to be dissipative with respect to a supply rate $w(u, y)$ if there exists a continuously differentiable storage function $V : \mathcal{R}^n \to \mathcal{R}$ that satisfies the dissipation inequality

$$\dot{V}(x) \leq w(u, y) \tag{6.6}$$

with $V(0) = 0$ and $V(x) > 0$ for all $x \neq 0$ where it is assumed that $x(t)$ is an implicit function of u and/or y as appropriate.[1] Following the work of [74,75] we are interested in supply rate functions that are quadratic in u and y. The general form of such supply rate functions is

$$w(u, y) = y^T Q y + 2u^T S y + u^T R u \tag{6.7}$$

where $Q \in \mathcal{S}^m$, $R \in \mathcal{S}^p$ and $S \in \mathcal{R}^{p \times m}$.

Definition 16 (Dissipativity) *The LTI system (6.5) is dissipative with respect to the supply rate function (6.7) if there exists a matrix P such that $V(x) = x^T P x$ where $P \succ 0$ and*

$$\begin{aligned} \dot{V}(x) &= x^T (A^T P + P A) x + u^T B^T P x + x^T P B u \\ &\leq w(u, y). \end{aligned}$$

Dissipativity can be verified using convex optimization by solving the Linear Matrix Inequality (LMI)

$$\begin{bmatrix} A^T P + P A - C^T Q C & P B - C^T S \\ B^T P - S^T C & -R \end{bmatrix} \preceq 0 \tag{6.8}$$

for $P \succ 0$. By altering the choices of the matrices (Q, S, R) different system

[1] Technically it is only required that $V(x) \geq 0$ for the system to be dissipative. However, by enforcing the strict inequality when $u = 0$ it is obvious that $V(x)$ is a Lyapunov function for the unforced system. As most of the results of this chapter are stability-based we will require positive definiteness of V.

properties can be ascertained as will be shown later. We will sometimes refer to a system as being (Q, S, R) dissipative for a particular choice of matrices.

One of the most useful properties of the (Q, S, R) dissipation framework is the manner in which networked systems can be analyzed. Consider the linear interconnection of M subsystems where each subsystem Σ_i is described by

$$\Sigma_i \begin{cases} \dot{x}_i = A_i x_i + B_i u_i \\ y_i = C_i x_i \end{cases} \tag{6.9}$$

and the interconnection (in negative feedback) between the subsystems is characterized by

$$u_i = -\sum_{i=1}^{M} H_{ij} y_j \quad \forall i. \tag{6.10}$$

Let us define $x = [x_1^T, \ldots, x_M^T]^T$, $u = [u_1^T, \ldots, u_M^T]^T$ and $y = [y_1^T, \ldots, y_M^T]^T$ then the interconnection can be written as $u = -Hy$ as in (6.4) and the full networked system dynamics given by

$$\dot{x} = Ax - BHCx \tag{6.11a}$$

$$y = Cx \tag{6.11b}$$

where $A = \text{diag}(A_1, \ldots, A_M)$, $B = \text{diag}(B_1, \ldots, B_M)$, and $C = \text{diag}(C_1, \ldots, C_M)$.

Remark 5 *Assume that for each subsystem Σ_i in the network there exists a matrix $P_i \succ 0$ such that Σ_i is dissipative with respect to a supply rate (Q_i, S_i, R_i). The inequality*

$$\sum_{i=1}^{M} \dot{V}_i(x_i) \leq \sum_{i=1}^{M} w_i(u_i, y_i) \tag{6.12}$$

holds regardless of the interconnection H.

Remark 6 *Define $P = diag(P_1, \ldots, P_M)$ where $P_i \succ 0$ and is of appropriate dimension. Let $V(x) = x^T P x$. We have that $\dot{V}(x) = x^T(A^T P + PA)x + u^T B^T P x +$*

$x^T PBu = \sum_{i=1}^{M} \dot{V}_i(x_i)$ *where* A, B *and* C *are as defined in (6.11). Furthermore* $w(u,y) = \sum_{i=1}^{M} w_i(u_i, y_i)$ *is given by (6.7) with* $Q = diag(Q_1, \ldots, Q_M)$, $S = diag(S_1, \ldots, S_M)$, $R = diag(R_1, \ldots, R_M)$.

Substituting the interconnection (6.10) into the inequality (6.12) produces

$$\sum_{i=1}^{M} \dot{V}_i(x_i) \leq y^T \widehat{Q} y$$

where

$$\widehat{Q} = Q - S^T H - H^T S + H^T R H,$$

which leads to the following sufficient condition for stability of the networked system:

Lemma 2 ([74]) *Assume each subsystem* Σ_i *is observable. The global interconnected system (6.11) is asymptotically stable if* $\widehat{Q} \prec 0$. *Furthermore,* $V(x) = \sum_{i=1}^{M} V_i(x_i) = x^T P x$ *is a Lyapunov function for the global system, with* $P = diag(P_1, \ldots, P_M)$.

Stability proofs such as the one described above will be derived for various system decompositions in Section 6.4. In particular we will examine the cases of passivity, dissipativity and bounded gain amongst others.

6.2.2 Small Gain Theorem

The small-gain theorem is a powerful tool for analyzing input-output stability of systems interconnected in feedback. In this section a very general version of the small-gain theorem is presented that relates finite-gain $\mathcal{L}$ stability[1] of two subsystems to the stability of the feedback connection when viewed as an input-output map. The version of the theorem used here is applicable to nonlinear

[1] With slight abuse of notation we sometimes refer to $\mathcal{L}_p$ stability as $\mathcal{L}$ stability when it does not matter which p-norm is used.

time-domain systems. For a more classical frequency domain interpretation see for example [17, 117].

The $\mathcal{L}_p$ space for $1 \leq p \leq \infty$ is defined as the set of all piecewise continuous functions $u : [0, \infty) \to \mathcal{R}^m$ such that the norm

$$\|u\|_{\mathcal{L}_p} = \left(\int_0^\infty \|u(t)\|^p dt \right)^{1/p} \tag{6.13}$$

is finite.

The extended space $\mathcal{L}_e$ is defined by

$$\mathcal{L}_e = \{u \mid u_\tau \in \mathcal{L},\ \forall \tau \in [0, \infty)\}$$

where u_τ is the truncation

$$u_\tau(t) = \begin{cases} u(t), & 0 \leq t \leq \tau \\ 0, & t > \tau \end{cases}.$$

The extended space allows us to consider unbounded signals that only belong to an $\mathcal{L}_p$ space on a finite time interval.

Input-output stability is concerned with determining if the feedback connection of two systems $G_1 : \mathcal{L}_e \to \mathcal{L}_e$ and $G_2 : \mathcal{L}_e \to \mathcal{L}_e$ (Figure 6.1) that are finite-gain stable, i.e.

$$\|y_{i_\tau}\|_{\mathcal{L}} \leq \gamma_1 \|e_{i_\tau}\|_{\mathcal{L}} + \beta_i, \quad \forall e_i \in \mathcal{L}_e, \quad \forall \tau \in [0, \infty), \quad i = 1, 2$$

produces a finite-gain $\mathcal{L}$ stable mapping $u \to y$ or $u \to e$ where $u \triangleq [u_1^T, u_2^T]^T$, $y \triangleq [y_1^T, y_2^T]^T$ and $e \triangleq [e_1^T, e_2^T]^T$. The small-gain theorem provides sufficient conditions for stability of the feedback connection shown in Figure 6.1

Theorem 10 ([4]) *Let γ_i denote the $\mathcal{L}_p$ norm of G_i. If G_1 and G_2 are both finite-gain $\mathcal{L}$ stable then the feedback connection is $\mathcal{L}$ stable if $\gamma_1 \gamma_2 < 1$.*

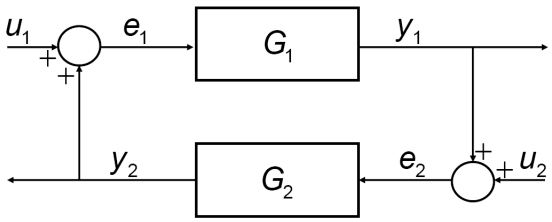


Figure 6.1: Small-gain feedback interconnection

In the context of system decomposition, the integration step is clearly less computationally demanding than the composite Lyapunov function approach as stability boils down to verifying that $\gamma_1\gamma_2 < 1$. In the following section we will describe how the $\mathcal{L}_2$ norm of a system can be computed using convex optimization and further, how this can be done using decomposition for systems with a large state dimension.

6.3 $\mathcal{L}_2 \to \mathcal{L}_2$ Performance via Decomposition

The $\mathcal{L}_2 \to \mathcal{L}_2$ gain of (6.1) is given by

$$\sup_{\|u\|_{\mathcal{L}_2}\neq 0} \frac{\|y\|_{\mathcal{L}_2}}{\|u\|_{\mathcal{L}_2}}, \quad x(0) = 0. \tag{6.14}$$

In general it is only possible to find an upper bound for the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of a nonlinear system on a bounded region of the state space. We will now outline two traditional methods for computing such an upper bound before proceeding to describe how these methods can be applied to larger systems via system decomposition.

Consider system (6.1) where $f(x)$ is locally Lipschitz and $G(x)$ and $h(x)$ are continuous on $\mathcal{R}^n$. Moreover, the matrix G has dimension $n \times p$ and $h : \mathcal{R}^n \to \mathcal{R}^m$. It is also assumed that $f(0) = 0$ and $g(0) = 0$.

Theorem 11 *[118] Consider the nonlinear time-invariant system given by (6.1). Let γ be a scalar such that $\gamma > 0$ and suppose there exists a positive semidefi-*

nite, continuously differentiable function $V(x)$ that satisfies the Hamilton-Jacobi Inequality:

$$\frac{\partial V}{\partial x} f(x) + \frac{1}{2\gamma^2} \frac{\partial V}{\partial x} G(x) G^T(x) \left(\frac{\partial V}{\partial x} \right)^T + \frac{1}{2} h^T(x) h(x) \leq 0 \tag{6.15}$$

for all $x \in \mathbb{R}^n$. For each x_0, system (6.1) is finite-gain $\mathcal{L}_2$ stable and its $\mathcal{L}_2$-gain is less than or equal to γ.

As can be seen from Theorem 11, even determining an upper bound on the $\mathcal{L}_2$ gain can be challenging for several reasons; i) computing $V(x)$ and ensuring it is positive semidefinite is not trivial for high order systems, and ii) the value of γ is linked to the choice of $V(x)$ thus obtaining tight bounds is difficult.

Finding solutions to the Hamilton-Jacobi inequality directly can be both analytically and computationally challenging, particularly when the system has a high state dimension. However, using dissipation inequalities and sum of squares programming we can compute bounds on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain. First we note that system (6.1) can be equivalently written as

$$\begin{aligned} \dot{x} &= f(x,u), \quad x(0) = x_0 \\ y &= h(x). \end{aligned} \tag{6.16}$$

The following proposition uses a Storage function argument to compute bounds on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of a system.

Proposition 3 *Given a system of the form (6.16) and a domain $\mathcal{D}$ that contains the origin, if there exists a continuously differentiable function $V : \mathcal{D} \to \mathcal{R}$ such that:*

1. $V(0) = 0$, $V(x) > 0 \ \forall x \in \mathcal{D}$.
2. $-\frac{\partial V}{\partial x} f(x,u) - y^T y + \gamma^2 u^T u \geq 0 \ \forall x \in \mathcal{D}$

then the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of (6.16) is less than or equal to γ for all $x_0 \in \mathcal{D}$.

Proof: Integrating Condition 2 from 0 to $T \geq 0$ and rearranging terms we obtain

$$\begin{aligned} V(x(T)) - V(x(0)) &\geq \int_0^T \left(y^T y - \gamma^2 u^T u\right) dt \\ &= \int_0^T y^T y dt - \gamma^2 \int_0^T u^T u dt. \end{aligned}$$

By definition $x(0) = 0$ and from Condition 1 we have $V(x(0)) = 0$ and $V(x(T)) \geq 0$, therefore $\int_0^T y^T y dt - \gamma^2 \int_0^T u^T u dt \leq 0$, giving

$$\frac{\sqrt{\int_0^T y^T y dt}}{\sqrt{\int_0^T u^T u dt}} \leq \gamma$$

for all $T \geq 0$. This is equivalent to (6.14). ■

Restricting our attention to the case where f and h are vectors of polynomial functions the following sum of squares programme derived from Proposition 3 can be used to compute bounds on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of (6.16):

Given system (6.1) and a positive definite function $\varphi(x)$, let the domain of interest $\mathcal{D}$ be described by a semi-algebraic set

$$\mathcal{D} \triangleq \{x \in \mathcal{R}^n \mid b_i(x) \leq 0, \quad i = 1, \ldots, k\} \tag{6.17}$$

where $b_i \in \mathcal{R}[x]$ for $i = 1, \ldots, k$. If a continuously differentiable polynomial function $V \in \mathcal{R}[x]$ and SOS polynomials $q_i \in \Sigma[x]$ for $i = 1, \ldots, k$ can be found that are solutions to:

$$\begin{aligned} \min_{\gamma > 0, V \in \mathcal{R}[x], q_i \in \Sigma[x]} \quad & \gamma \\ \text{s.t.} \quad & V(x) - \varphi(x) \in \Sigma[x] \\ & -\frac{\partial V}{\partial x} f(x,u) - y^T y + \gamma u^T u + \sum_{i=1}^{k} q_i(x) b_i(x) \in \Sigma[x] \end{aligned} \tag{6.18}$$

then the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of (6.16) is upper bounded by $\sqrt{\gamma}$.

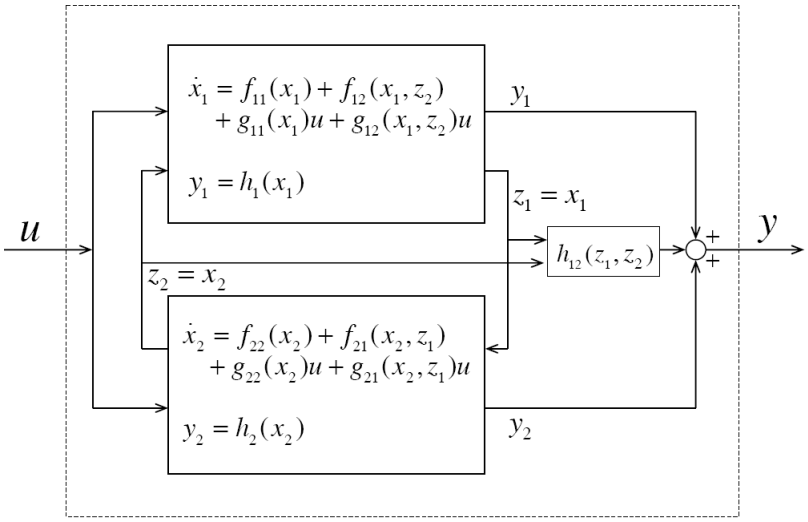


Figure 6.2: Block diagram of the decomposed non-autonomous dynamical system described by (6.19).

As was shown in previous chapters, the task of computing V and finding sum of squares polynomials g_i that solve the optimization problem above is computationally demanding and simply not practical for systems of high dimension. In this case a decomposition approach can be used to help with scalability. The decomposition algorithm from Section 4.4.2 can be applied to the drift dynamics of system (6.1). Focussing on the case of $M = 2$ the decomposition of (6.1) is

$$\begin{aligned} \dot{x}_1 &= f_{11}(x_1) + f_{12}(x_1, z_2) + g_{11}(x_1)u + g_{12}(x_1, z_2)u \\ \dot{x}_2 &= f_{22}(x_2) + f_{21}(x_2, z_1) + g_{22}(x_2)u + g_{21}(x_2, z_1)u \\ z_1 &= x_1 \\ z_2 &= x_2 \\ y &= h_1(x_1) + h_2(x_2) + h_{12}(x_1, x_2) \end{aligned} \tag{6.19}$$

where u is the external input, y is the external output, $h_{12}(x_1, x_2) = h(x) - h_1(x_1) - h_2(x_2)$ and z_1, z_2 provide the internal system coupling. A block diagram representation of (6.19) is shown in Figure 6.2.

Clearly trying to analyze (6.19) will incur the same computational cost as

analyzing (6.1) directly. If we apply the decomposition algorithm from the previous chapter on (6.1) with $u = 0$, $y = 0$ then the effect of f_{12}, f_{21} on (6.19) is minimized. If in addition we ignore the coupling at the input and output then we are left with the decoupled systems

$$\dot{x}_1 = f_{11}(x_1) + g_{11}(x_1)u, \quad y_1 = h_1(x_1), \tag{6.20a}$$

$$\dot{x}_2 = f_{22}(x_2) + g_{22}(x_2)u, \quad y_2 = h_2(x_2). \tag{6.20b}$$

For systems of this form, Proposition 3 tells us how to construct Storage functions for the decoupled systems. In almost exactly the same manner as in Chapter 4 we define the composite Storage function as $V_c = \sum_{i=1}^{M} \alpha_i V_i(x_i)$. The decomposition algorithm for the non-autonomous system is as follows:

1. Apply the decomposition algorithm from Section 4.4.2 to the drift dynamics of (6.1), i.e. $f(x)$.

2. Use the resulting state partition to form (6.19) and hence the decoupled subsystems (6.20a–6.20b).

3. Compute storage functions V_i for subsystem (6.20a) and (6.20b) using the sum of squares programme (6.18).

4. Use sum of squares programming to find positive scalars $\alpha_1, \alpha_2, \zeta$ and SOS polynomials $r_i \in \Sigma[x]$ for $i = 1, \ldots, k$ that solve:

$$\begin{aligned} \min_{\{\alpha_i, \zeta\} > 0, r_i \in \Sigma[x]} \quad & \zeta \\ \text{s.t.} \quad & \zeta > 0, \alpha_1 > 0, \alpha_2 > 0 \\ & -\frac{\partial V_c}{\partial x} f(x,u) - y^T y + \zeta u^T u + \sum_{i=1}^{k} b_i(x) r_i(x) \in \Sigma[x] \end{aligned}$$

   where $V_c = \alpha_1 V_1(x_1) + \alpha_2 V_2(x_2)$ and the polynomials $b_i(x)$ describe $\mathcal{D}$.

5. An upperbound on the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of system (6.1) is then given by $\sqrt{\zeta}$.

It is possible that further computational savings may be achieved if the SOS polynomials q_i in (6.18) corresponding to the domain restriction in the subproblems are used in place of the SOS polynomials r_i is Step 4. However in many instances such an approach may be overly restrictive.

6.3.1 Input-output Numerical Example

In this example we consider a fourth order system with a quartic vector field:

$$\begin{aligned}
\dot{x}_1 &= -3x_1 - x_1x_2^2 + x_1x_4 + 2x_1x_4u - x_1x_2u \\
\dot{x}_2 &= -2x_1^2 + 0.5x_1 - x_1^2x_2^2 - 3x_2 - 4x_1x_3u \\
\dot{x}_3 &= -x_3^4 - x_3^2x_4^2 + 1.5x_4 - x_3 - x_4u \\
\dot{x}_4 &= -x_4^2 - 4x_3 - 2x_1 + x_3x_4u \\
y_1 &= x_1 + x_2 \\
y_2 &= x_4
\end{aligned} \tag{6.21}$$

This system is sufficiently small to be analyzed directly allowing us to make a comparison between gain estimates and the size of the optimization problems. Sum of squares programming was first used to construct a quadratic Lyapunov function that certifies local asymptotic stability of (6.21) with $u = 0$. The Lyapunov function found is

$$\begin{aligned}
V(x) &= 0.0043x_1^2 + 0.0006x_1x_2 + 0.0065x_2^2 + 0.0118x_1x_3 - 0.0012x_2x_3 \\
&\quad +0.0384x_3^2 - 0.006x_1x_4 + 0.0016x_2x_4 - 0.0118x_3x_4 + 0.0199x_4^2.
\end{aligned}$$

Implementing the sum of squares optimization (6.18) we obtain an upper bound on the $\mathcal{L}_2$ gain of (6.21) of 0.0907 (to 4dp). Applying the decomposition algorithm to (6.21) with $u = 0$, the partitions $\mathcal{T}_1 = \{x_1, x_2\}$, $\mathcal{T}_2 = \{x_3, x_4\}$ are obtained

resulting in the following decomposition:

$$f_{11} = \begin{pmatrix} -3x_1 - x_1x_2^2 \\ -2x_1^2 + 0.5x_1 - x_1^2x_2^2 - 3x_2 \end{pmatrix}$$

$$f_{12} = \begin{pmatrix} x_1x_4 \\ 0 \end{pmatrix} \quad f_{21} = \begin{pmatrix} 0 \\ -2x_1 \end{pmatrix}$$

$$g_{11} = \begin{pmatrix} -x_1x_2 \\ 0 \end{pmatrix} \quad g_{12} = \begin{pmatrix} 2x_1x_4 \\ -4x_1x_3 \end{pmatrix}$$

$$f_{22} = \begin{pmatrix} -x_3^4 - x_4^2x_3^2 + 1.5x_4 - x_3 \\ -x_4^2 - 4x_3 \end{pmatrix}$$

$$g_{21} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad g_{22} = \begin{pmatrix} -x_4 \\ x_3x_4 \end{pmatrix}$$

$$h_1 = x_1 + x_2 \quad h_2 = x_4 \quad h_{12} = 0.$$

Storage functions V_1 and V_2 corresponding to the decoupled subsystems $S_1(\mathcal{T}_1)$ and $S_2(\mathcal{T}_2)$ (6.20a–6.20b) are computed by solving (6.18) where:

$$\begin{aligned} V_1 &= 1.427x_2^2 + 0.018x_1x_2 + 0.503x_1^2 \\ V_2 &= 1.168x_4^2 - 1.037x_3x_4 + 3.113x_3^2. \end{aligned}$$

The multipliers $\alpha_1 = 2.1505$ and $\alpha_2 = 1$ are found using Step 4 in the algorithm described above that produce a composite Storage function $V_c = \alpha_1 V_1 + \alpha_2 V_2$ for system (6.21) with an upper $\mathcal{L}_2 \to \mathcal{L}_2$ gain bound of 0.1711 (to 4dp). This bound can be tightened at the expense of searching for higher order Storage functions, keeping V_1 and using a quartic Storage function V_2 a bound of 0.122 is found. This example illustrates that it is possible to achieve $\mathcal{L}_2 \to \mathcal{L}_2$ gain bounds using a decomposition approach. The following example illustrates just how useful the method can be in practice. Consider the system:

$$\begin{aligned}
\dot{x}_1 &= -2x_1 + 0.5x_1x_2 - 3x_3^2 - x_6^2 + u \\
\dot{x}_2 &= x_1 - 3x_2 - 4x_2x_3 + x_6 + 2x_2u \\
\dot{x}_3 &= -x_1x_2 - 2x_2 - 3x_3 + x_1x_2x_3 - 2x_4 \\
\dot{x}_4 &= -x_5 - 3x_4 - x_4x_5x_6 + 10x_5^2 - x_2 \\
\dot{x}_5 &= -2x_5 - 3x_6^3 + x_4x_5x_3^2 - x_5x_6u \\
\dot{x}_6 &= -4x_6 + x_4x_6 + x_5^2x_6 + x_1x_6u \\
y_1 &= x_1 + x_2 \\
y_2 &= x_4^2 \\
y_3 &= -x_2x_6
\end{aligned} \tag{6.22}$$

In this instance using direct methods it was not possible to construct Storage function using quadratic polynomials. Using quartic polynomials a storage function was constructed although there was a large residual error present, the computation took 55s and produced a gain bound of 1.537. Applying the decomposition algorithm as described previously produces the state partitions $\mathcal{T}_1 = \{x_1, x_2, x_3\}$ and $\mathcal{T}_2 = \{x_4, x_5, x_6\}$. Using a quartic storage function for the system $S(\mathcal{T}_1)$ and a quadratic storage function for $S(\mathcal{T}_2)$ a gain bound of 1.414 was achieved in 32s. This time includes the time taken to construct the two storage functions and to search for the multipliers to verify the derivative condition of the composite

Storage function and compute the bound. The Storage functions and multipliers are given in the Appendix A.1 at the end of the chapter.

6.4 Interconnected Systems

In this section we describe some stability results based on composite methods that verify the stability of the original system through subsystem Storage functions. We assume that a decomposition algorithm has already been applied, and the subsystems created. In Section 6.5 a clustering algorithm is proposed that aims to take into account these stability metrics.

Consider the LTI system G described by (6.2) with the state vector $\hat{x} \in \mathcal{R}^n$. Now assume that it has been decomposed into two subsystems G_1 and G_2 connected in feedback as in (6.3) where the state vector has been permuted such that $x \triangleq [x_1^T, x_2^T]^T = T\hat{x}$, where $T \in \Pi_n$ and $x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}$, $n_1 + n_2 = n$ and no state belongs to multiple subsystems.

Proposition 4 *Assume G has been decomposed into G_1 and G_2 which are dissipative with respect to the quadratic supply rates $w_1(u_1, y_1)$ and $w_2(u_2, y_2)$ of the form (6.7) respectively. If $w_1(u_1, y_1) + w_2(u_2, y_2) < 0$ for all input output pairs then the sum of the Storage functions $V_1(x_1) + V_2(x_2)$ is a Lyapunov function that verifies that the equilibrium point of (6.2) is asymptotically stable.*

Proof: Direct application of Lemma 2. ■

Note that this and all further results generalize to the case where G can be decomposed into multiple subsystems. For the sake of clarity we focus here on the case of two interacting subsystems.

If we assume an even more generic subsystem model than that of the interconnection structure for G_1, G_2 of the form (6.4)[1] then the right hand side of

[1]Note the difference in sign between (6.4) and (6.10), this is due to the decomposition providing systems coupled in positive rather than negative feedback. It is trivial to re-write the decomposition to obtain a negative feedback connection.

$\dot{V}_1(x_1) + \dot{V}_2(x_2) \leq w_1(u_1, y_1) + w_2(u_2, y_2)$ with (u_i, y_i) obtained from the decomposition is given by

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} (H_{11}^T R_1 H_{11} + H_{11}^T S_1 + S_1^T H_{11} + \\ + H_{21}^T R_2 H_{21} + Q_1) & (H_{11}^T R_1 H_{12} + S_1^T H_{12} + \\ \star & + H_{21}^T S_2 + H_{21}^T R_2 H_{22}) \\ & (H_{22}^T R_2 H_{22} + H_{22}^T S_2 + S_2^T H_{22} + \\ & + H_{12}^T R_1 H_{12} + Q_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{6.23}$$

This expression is derived by setting $y_i = x_i$ in (6.4) and writing $w_1(u_1, y_1) + w_2(u_2, y_2)$ in matrix form.

One interpretation of dynamical system decomposition is to determine the form of the interconnection matrix H in order to make (6.23) negative definite. Note that (6.23) corresponds to G_i in the form of (6.3) and not the more general case of Σ_i as in (6.9). The results extend easily to the more general case, however this complicates the notation and so in the interest of clarity we do not pursue this further.

By appropriate choice of (Q, S, R) matrices, the supply functions (6.7) can represent passivity, finite-gain and dissipativity, each of which alters the structure of (6.23). The remainder of this section examines each of these cases in turn and provides stability tests for (6.2) based on its decomposed subsystems (6.3).

6.4.1 Passivity

An LTI system of the form (6.5) is said to be passive if it is dissipative with respect to supply rate (6.7) with $Q_i = 0, S_i = I, R_i = 0$ (which from now on we denote as $(0, I, 0)$-dissipative) and LMI (6.8) is feasible.

Equivalently (6.5) is passive if for every trajectory with $x(0) = 0$ the integral inequlity

$$\int_0^T u^T y \; dt \geq 0$$

holds for all $T \geq 0$.

Assume that G has been decomposed into G_1, G_2 and that G_i is $(0, I, 0)$-dissipative. Substituting these matrices into the supply rate function (6.7) and applying the feedback interconnection (6.4) we see from Equation (6.23) that we require

$$\begin{bmatrix} H_{11}^T S_1 + S_1^T H_{11} & S_1^T H_{12} + H_{21}^T S_2 \\ \star & H_{22}^T S_2 + S_2^T H_{22} \end{bmatrix} \prec 0 \tag{6.24}$$

for (asymptotic) Lyapunov stability,[1] where from the definition of G_i (6.3) we have that

$$H = \begin{bmatrix} 0 & A_{12} \\ A_{21} & 0 \end{bmatrix}.$$

With this interconnection structure and system decomposition the diagonal block entries in (6.24) are zero and the off diagonal blocks are given by $A_{12} + A_{21}^T$ and its transposition. In this form (6.24) cannot be negative definite as its eigenvalues will be real and symmetric about the imaginary axis. This problem can be alleviated if we consider a slight modification to the decomposition described by (6.3) by imposing a further decomposition on the drift matrices A_{ii} and including a feedback term. The new decomposition for G_1 is

$$\widehat{G}_1 \begin{cases} \dot{x}_1 &= \epsilon_1 A_{11} x_1 + u_1 \\ u_1 &= A_{12} x_2 + \delta_1 A_{11} x_1 \\ y_1 &= x_1 \end{cases} \tag{6.25}$$

where $\epsilon_1 + \delta_1 = 1$ and we assume all matrices are of compatible dimension. In the same manner $\widehat{G}_2$ can be constructed. The LMI (6.24) is then replaced by

$$\begin{bmatrix} \delta_1(A_{11} + A_{11}^T) & A_{12} + A_{21}^T \\ \star & \delta_2(A_{22} + A_{22}^T) \end{bmatrix} \prec 0. \tag{6.26}$$

When $\widehat{G}_1, \widehat{G}_2$ are dissipative with respect to $(0, I, 0)$ and LMI (6.26) is feasible

[1] Of course non-asymptotic Lyapunov stability may still be achievable.

the original system (6.2) is stable as verified by the Lyapunov function $V(x) = V_1(x_1) + V_2(x_2)$.

An alternative approach is to select ϵ_i, δ_i arbitrarily (ensuring $\epsilon_1 + \delta_1 = 1$) and using the modified decomposition $\widehat{G}_i$ solve LMI (6.24) where the decision variables are the diagonal matrices $S_i \succ 0$. Such an approach is possible because any system that is dissipative with respect to $(0, I, 0)$ is also dissipative with respect to any $(0, X, 0)$ supply rate with X diagonal and $X \succ 0$.

6.4.2 Finite Gain

For LTI systems of the form (6.5) the $\mathcal{L}_2 \to \mathcal{L}_2$ gain can be calculated by solving:

$$\begin{aligned} \min_{\gamma \in \mathcal{R}, P \in \mathcal{S}} \quad & \gamma \\ \text{s.t.} \quad & \begin{bmatrix} A^T P + P A + C^T C & PB \\ B^T P & -\gamma I \end{bmatrix} \preceq 0 \\ & P \succ 0, \quad \gamma > 0. \end{aligned} \tag{6.27}$$

The $\mathcal{L}_2 \to \mathcal{L}_2$ gain is then given by $\sqrt{\gamma}$ [5]. This is a direct application of the KYP lemma (see Appendix A.2 of this chapter).

Provided that the LTI system is minimal, i.e. (A, C) is observable and (A, B) is controllable, the $\mathcal{L}_2 \to \mathcal{L}_2$ gain is equivalent to the $\mathcal{H}_\infty$-norm[1] of the frequency domain realization of (6.5) given by $F(s) = C(sI - A)^{-1}B$. In this instance γ obtained from solving (6.27) is exact.

Given two systems connected in feedback, the small-gain theorem (see Section 6.2.2) states that if $\gamma_1 \gamma_2 < 1$ then the feedback connection is stable. A generalization of the small gain theorem for networks is given in [119]. Following from LMI (6.27) it can be seen that the supply rate functions associated with finite gain analysis are given by $(-I, 0, \gamma_i^2 I)$ for $i = 1, 2$. Following a similar procedure

[1] The $\mathcal{H}_\infty$-norm of the system $F(s)$ is defined as $\|F\|_{\mathcal{H}_\infty} \triangleq \sup_{Re(s)>0} \bar{\sigma}[F(s)] = \sup_{\omega \in \mathcal{R}} \bar{\sigma}[F(j\omega)]$.

as in the previous section we obtain the following stability requirement.

Substituting the appropriate Q, S, R matrices and interconnection structure (assuming no additional feedback) into (6.23) produces the following stability requirement:

Lemma 3 *Assume system (6.2) has been decomposed into the subsystems given in (6.3). Further assume that the subsystems are dissipative with respect to $S_i = 0, Q_i = -I$ and $R_i = \gamma_i^2 I$ where γ_i denotes the $\mathcal{L}_2 \to \mathcal{L}_2$-norm of subsystem i. Then if*

$$max\ \{\gamma_2 \bar{\sigma}(A_{21}), \gamma_1 \bar{\sigma}(A_{12})\} < 1$$

system (6.2) is asymptotically stable as verified by the Lyapunov function $V_1(x_1) + V_2(x_2)$.

Proof: Substituting the appropriate Q, S, R matrices into (6.23) gives

$$\begin{aligned} &\begin{bmatrix} \gamma_2^2 A_{21}^T A_{21} & 0 \\ 0 & \gamma_1^2 A_{12}^T A_{12} \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \prec 0 \\ \Leftrightarrow\ &\bar{\sigma}\left(\begin{bmatrix} \gamma_2 I & 0 \\ 0 & \gamma_1 I \end{bmatrix} \begin{bmatrix} A_{21} & 0 \\ 0 & A_{12} \end{bmatrix} \right) < 1 \end{aligned} \tag{6.28}$$

which is equivalent to the condition in the theorem. ■

What is useful about this Lemma is that it provides a decentralized stability check. Stability of the interconnected system can be verified by testing if each subsystem satisfies $\gamma_i \bar{\sigma}(A_{ij}) < 1$, it is not necessary to consider the whole system.

Lemma 3 and Equation (6.28) provide a *nominal* stability test for the decomposed subsystems. What would be desirable is to determine the maximum $\mathcal{L}_2 \to \mathcal{L}_2$ gains (i.e. γ_i's) such that (6.28) holds. Such a characterization would provide a robustness measure for the decomposed system. In addition it will provide a measure of how good the decomposition is. From the equivalence relation in (6.28) the maximum achievable γ's denoted by $\hat{\gamma}$ that satisfy the stability

requirement in Lemma 3 are given by $\hat{\gamma}_1 = \bar{\sigma}(A_{12})^{-1}$ and $\hat{\gamma}_2 = \bar{\sigma}(A_{21})^{-1}$.

If we consider more generic supply rates with $(-\kappa I, 0, \kappa\gamma^2 I)$, $\kappa > 0$ instead of $(-I, 0, \gamma^2 I)$ then it is possible to strengthen Lemma 3. Observe that if a system is dissipative with respect to $(-I, 0, \gamma^2 I)$ then it is also dissipative with respect to $(-\kappa I, 0, \kappa\gamma^2 I)$ for any $\kappa > 0$. Taking this into consideration we obtain the following result:

Lemma 4 *If there exists a scalar $\kappa > 0$ such that G_1 and G_2 are dissipative w.r.t. $(-\kappa I, 0, \kappa\gamma_1^2 I)$ and $(-\kappa I, 0, \kappa\gamma_2^2 I)$ respectively then system (6.2) is stable if max $\{\kappa\gamma_1\bar{\sigma}(A_{21}), \kappa^{-1}\gamma_2\bar{\sigma}(A_{12})\} < 1$.*

Note that when $\bar{\sigma}(A_{21}) = \bar{\sigma}(A_{12})$, Lemma 4 is equivalent to the small gain condition. The following proposition based on simple dissipation inequality arguments relates small-gain stability to the existence of a composite Lyapunov function.

Proposition 5 *Given two observable subsystems G_1 and G_2 of the form (6.3) connected in feedback such that the small-gain theorem is satisfied, then there always exists an $\alpha > 0$ such that a composite quadratic Lyapunov function of the form $V_c(x) = V_1(x_1) + \alpha V_2(x_2)$ exists that verifies the stability of the original system (6.2).*

Proof: In order for the subsystems G_i to satisfy Theorem 10 they must both be $\mathcal{L}_2 \to \mathcal{L}_2$ stable, thus there exist storage functions that satisfy

$$\begin{aligned} V(0) &= 0 \\ V_i(x_i) &> 0 \quad (\forall x \neq 0) \\ \dot{V}_i(x_i) &\leq \gamma_i^2\|u_i\|^2 - \|y_i\|^2 \quad \gamma_i > 0 \end{aligned}$$

for $i = 1, 2$. Clearly $V_c > 0$ by definition, the derivative condition is now given by

$$\dot{V}_1(x_1) + \alpha\dot{V}_2(x_2) \leq \gamma_1^2\|u_1\|^2 - \|y_1\|^2 + \alpha(\gamma_2^2\|u_2\|^2 - \|y_2\|^2).$$

Substituting in the feedback connection $y_1 = u_2$ and $y_2 = u_1$ the right hand side of the inequality above becomes

$$(\gamma_1^2 - \alpha)\|u_1\|^2 + (\alpha\gamma_2^2 - 1)\|u_2\|^2$$

which is negative only when $\gamma_1^2 - \alpha < 0$ and $\alpha < 1/\gamma_2^2$ which is only achievable when $\gamma_1^2\gamma_2^2 < 1$ which is the small-gain theorem requirement. ■

6.4.3 Input Strong Passivity

A system of the form (6.5) is said to be *input strongly passive* with a dissipation rate $\eta < 0$ if

$$\int_0^T u^T y - \eta u^T u \; dt \geq 0, \quad x(0) = 0$$

for all $T \geq 0$ [5]. The maximum dissipation achievable is the largest η such that the integral inequality above holds. This can be computed by solving the SDP:

$$\begin{aligned} \max \quad & \eta \\ \text{s.t.} \quad & \begin{bmatrix} A^T P + PA & PB - C^T \\ B^T P - C & 2\eta I \end{bmatrix} \prec 0 \\ & P \succ 0, \quad \eta < 0. \end{aligned} \tag{6.29}$$

Setting $\tau = -2\eta$ we obtain a Q, S, R supply rate of $(0, I, -\tau I)$. Substituting the appropriate supply rate functions $(0, I, -\tau_i I)$ corresponding to G_1 and G_2 into Equation (6.23) the requirement for $V_1(x_1) + V_2(x_2)$ to be a composite Lyapunov function for (6.2) becomes

$$\begin{bmatrix} \tau_2 A_{21}^T A_{21} & A_{12} + A_{21}^T \\ A_{12}^T + A_{21} & \tau_1 A_{12}^T A_{12} \end{bmatrix} \prec 0 \tag{6.30}$$

when $H_{11} = H_{22} = 0$. The LMI (6.30) is never feasible as the elements on the diagonal will always be positive definite. However if we allow for a feedback term in each subsystem by modifying the decomposition according to (6.25) then it is possible to make (6.23) negative definite. The modified LMI is easily obtained from (6.23) with the appropriate substitutions: First factor (6.30) as follows:

$$\begin{bmatrix} A_{21} & 0 \\ 0 & A_{12} \end{bmatrix}^T \begin{bmatrix} \tau_2 & 0 \\ 0 & \tau_1 \end{bmatrix} \begin{bmatrix} A_{21} & 0 \\ 0 & A_{12} \end{bmatrix} + \begin{bmatrix} 0 & A_{21}^T + A_{12} \\ A_{12}^T + A_{21} & 0 \end{bmatrix} \succ 0.$$

Substitute into this the feedback terms from (6.25), then A_{12} and A_{21} in (6.31) become $A_{12} + \delta_1 A_{11}$ and $A_{21} + \delta_2 A_{22}$ respectively[1] and equation (6.31) can be written as

$$\begin{bmatrix} A_{21} & 0 \\ 0 & A_{12} \end{bmatrix}^T \begin{bmatrix} \tau_2 & 0 \\ 0 & \tau_1 \end{bmatrix} \begin{bmatrix} A_{21} & 0 \\ 0 & A_{12} \end{bmatrix} + \begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix}^T \begin{bmatrix} \tau_2\delta_2^2 & 0 \\ 0 & \tau_1\delta_1^2 \end{bmatrix} \begin{bmatrix} A_{22} & 0 \\ 0 & A_{11} \end{bmatrix}$$
$$\prec - \begin{bmatrix} \tau_2\delta_2(A_{21}^T A_{22} + A_{22}^T A_{21}) & A_{21}^T + \delta_2 A_{22}^T + A_{12} + \delta_1 A_{11} \\ \star & \tau_1\delta_1(A_{12}^T A_{11} + A_{11}^T A_{12}) \end{bmatrix}.$$

Thus if δ_1, δ_2 can be found such that the LMI above is satisfied then both systems are strictly input passive and a composite Lyapunov function exists. Note that a simpler to implement heuristic is obtained by noting that the left hand side of the LMI above could be replaced by a maximum singular value expression

$$\bar{\sigma}\left(\begin{bmatrix} \sqrt{\tau_2}A_{21} & 0 \\ 0 & \sqrt{\tau_1}A_{12} \end{bmatrix}\right) + \bar{\sigma}\left(\begin{bmatrix} \sqrt{\tau_2}\delta_2 A_{22} & 0 \\ 0 & \sqrt{\tau_1}\delta_1 A_{11} \end{bmatrix}\right)$$

that we seek to minimize.

[1] Note that this does not affect the values τ_1, τ_2.

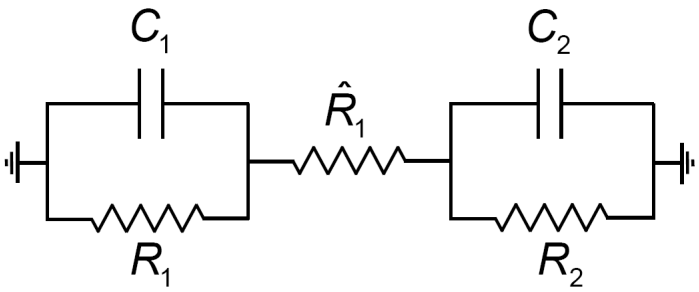


Figure 6.3: Two vertices corresponding to a resistor and capacitor, C_i, R_i connected via an edge resistor $\widehat{R}_1$ considered in the example in Section 6.4.4.

6.4.4 RC Network Example

Consider a resistor capacitor network that is represented by a graph with n nodes where each node consists of a resistor R_i connected in parallel to a capacitor C_i whic is grounded. Nodes are connected to each other via a resistor $\widehat{R}_i$. A 2-node example is illustrated in Figure 6.3.

Let all capacitors have unit capacitance, denote $x_i \in \mathcal{R}$ the voltage across capacitor C_i and $u_i \in \mathcal{R}$ the current entering node i. Each node may be modeled as the first order system

$$\begin{aligned} \dot{x}_i &= -g_i x_i + u_i \\ y_i &= x_i \end{aligned} \tag{6.31}$$

for $i = 1, \ldots, n$ where $g_i = R_i^{-1}$. The network can then be represented as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{Z})$ where the nodes $\mathcal{V}$ are represented by the states x_i given by (6.31), the topology is defined by the presence of a resistor connecting a pair of nodes and the edge weight $z_i \in \mathcal{Z}$ is given by $\widehat{R}_k^{-1}$. From this graph the weighted Laplacian $\mathcal{L}(\mathcal{G})$ can be constructed. Finally, define $G = \text{diag}(g_1, \ldots, g_n)$. The full networked dynamical system can now be compactly written as

$$\begin{aligned} \dot{x} &= -(\mathcal{L}(\mathcal{G}) + G)x \\ y &= x. \end{aligned} \tag{6.32}$$

Consider the RC network (6.32) with the matrix $-(\mathcal{L}+G) \triangleq \widehat{A}$:

$$\widehat{A} = \begin{bmatrix} -3.5176 & 0 & 0 & 0.4096 & 0 & 2.2119 \\ 0 & -2.0196 & 0 & 0 & 0.0353 & 0 \\ 0 & 0 & -3.5349 & 0 & 0 & 2.6817 \\ 0.4096 & 0 & 0 & -2.4146 & 0.5974 & 0 \\ 0 & 0.0353 & 0 & 0.5974 & -0.8270 & 0 \\ 2.2119 & 0 & 2.6817 & 0 & 0 & -7.8586 \end{bmatrix}.$$

Let $\mathcal{T}_1 = \{x_1, x_3, x_6\}$ and $\mathcal{T}_2 = \{x_2, x_4, x_5\}$ describe the system decomposition. Using LMI (6.27) to compute the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of the subsystems $S(\mathcal{T}_1)$ and $S(\mathcal{T}_2)$ we obtain $\gamma_1 = 0.1694$, $\gamma_2 = 0.2549$ respectively. The interconnection of these subsystems satisfy both the small gain theorem and Lemma 3. In addition these subsystems satisfy the stability criteria given by the passivity LMI derived in (6.26) with $\delta_1 = 0.42$, $\delta_2 = 1.1208$.

This system decomposition was found by iterating through all permutation matrices $T \in \Pi$ of order n. We observe that this is the only decomposition that satisfies the small gain condition. Clearly such an approach is not feasible for large systems. While system (6.32) has quite a sparse structure this is certainly not a requirement for applying this technique; in [15] the same system model with a fully connected network is successfully analyzed using a decomposition and dissipation inequality approach.

In the remainder of this chapter an algorithmic method for system decomposition is presented that can incorporate the ideas presented above.

6.5 A Clustering Based Decomposition Algorithm

In this section a different approach to system decomposition than the one taken in Chapter 4 is developed. Instead of a *top-down* approach to system decomposition which starts with a large-scale system and decomposes it into smaller subsystems,

here we consider a clustering based technique which can be thought of as a *bottom-up* apprach.

Let each state in the system $\dot{x} = Ax$ represent a node in a directed graph. Inputs and outputs of nodes are specifically considered. Each node can be thought of as a system with dynamics given by

$$\dot{x}_i = A_{ii}x_i + B_i u_i, \quad y_i = x_i$$

where B_i corresponds to the non-zero off-diagonal elements of the i^{th} row of A and the input vector corresponds to the appropriate states that $\dot{x}_i$ is a function of. The idea is conceptually very simple, find groups of states (clusters) that when lumped together form subsystems whose feedback interconnection forms a stable system. A similar manual technique referred to as lumping has been used in the context of structured model reduction [120]. Here, we would like to automate the process of decomposition and try to avoid an exhaustive combinatorial search.

6.5.1 Appending states to LTI systems

Previous work has looked at decomposing a dynamical system based on edge energy minimization. This involved computing the $\mathcal{L}_2$ norm of the output signal. Here we derive a bound on the maximum $\mathcal{L}_2$ norm of the output (referred to hereafter as the $\mathcal{L}_2$ energy) achievable when appending an additional state to an LTI system. It is envisioned that such bounds will be useful in a variety of situations such as when there is a high communication overhead, limited point to point communication or networks of systems that are of sufficient size and complexity that distributed algorithms become necessary.

An illustrative example is given below that considers the case of appending a single state system to a two state system in various configurations. It is trivial to extend this example to the general case and all the theoretical results follow through. For clarity all results are presented for the specific case. We consider

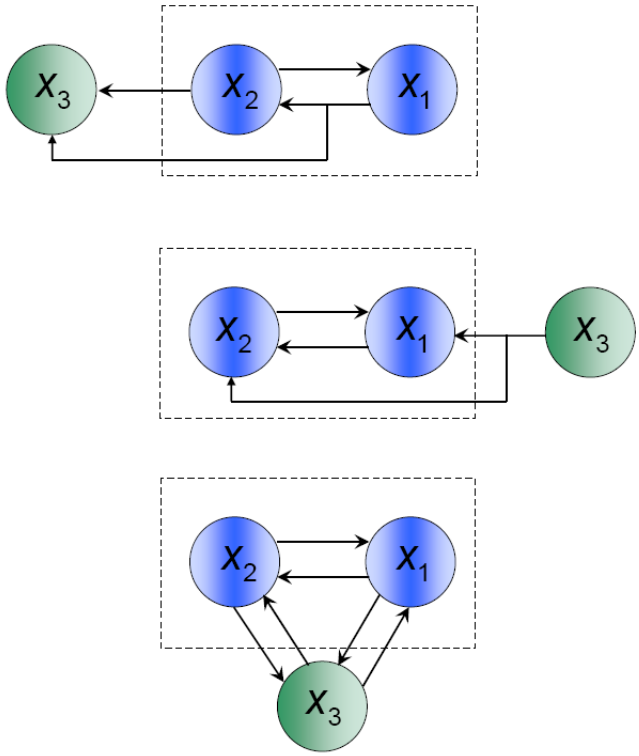


Figure 6.4: States can be appended to subsystems in one of three ways. Each node represents a state, blue nodes contained in dashed boxes represent the existing subsystem (6.33), the green node represents the state to be appended. In the first instance the new state is appended downstream of the existing subsystem (Top). This corresponds to $B_3 = 0$, $B_4 = 0$ in (6.34). Next we consider the case where the new state is appended up-stream of the existing subsystem (Middle). This corresponds to $B_1 = 0$, $B_2 = 0$ in (6.34). The final case corresponds to the new state fully interacting with the existing subsystem, i.e. $B_i \neq 0$ for $i = 1, \ldots, 4$.

instances where states can be appended up-stream, down-stream and in feedback as illustrated in Figure 6.4.

6.5.1.1 Lower bound estimates

Consider the 2-state LTI system

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}}_{z}; \quad \underbrace{\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}}_{v} = \underbrace{\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}}_{C} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (6.33)$$

where $z(0) = z_0$. We are concerned with finding a bound on the maximum $\mathcal{L}_2$ norm of the output of (6.33) when an additional state is appended, i.e. when the system becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_3 \\ A_{21} & A_{22} & B_4 \\ \hline B_1 & B_2 & A_{33} \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{6.34a}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \left[\begin{array}{cc|c} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ \hline 0 & 0 & C_{33} \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{6.34b}$$

where $x = [x_1, x_2, x_3]^T$, $y = [y_1, y_2, y_3]^T$ and the initial condition is $x(0) = x_0$. The upper left blocks in (6.34a) and (6.34b) correspond to the A and C matrices defined in (6.33).

To find a bound on max $\|y\|_{\mathcal{L}_2}$ with $\|x_0\|_2 = 1$ we will use bounds from the individual system components in a decentralized manner. We begin by presenting the case when $B_3 = 0, B_4 = 0$, and $C_{33} \in \mathcal{R}$, before generalizing the results. In this instance the extra state can be thought of as a *down stream* addition to the system. This is illustrated in Figure 6.4 (Top).

Theorem 12 *Define $\bar{C} = diag(C, C_{33})$. Assume that $B_3 = 0, B_4 = 0$, and $C_{33} \in \mathcal{R}$ and that A, A_{33} are Hurwitz matrices. Then*

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \geq \frac{1}{2}\|\bar{C}^T\bar{C}\|_2 \left[max\{\overline{\sigma}(A), A_{33}\} + \overline{\sigma}(\widetilde{B})\right]^{-1} \tag{6.35}$$

where

$$\widetilde{B} = \begin{bmatrix} B_1^T B_1 & B_1^T B_2 & B_1^T A_{33} \\ B_2^T B_1 & B_2^T B_2 & B_2^T A_{33} \\ A_{33}^T B_1 & A_{33}^T B_2 & 0 \end{bmatrix}. \tag{6.36}$$

Proof: First consider an LTI system such as (6.33), then $\|v\|_{\mathcal{L}_2}^2 = z_0^T P z_0$

where $P \succeq 0$ solves $A^T P + PA = -C^T C$, the maximum is given by $\bar{\lambda}(P)$ and the initial condition that achieves this is the corresponding (unit) eigenvector.

Denote the system matrix in (6.34) by $\bar{A}$, then

$$\begin{aligned} \|\bar{C}^T\bar{C}\|_2 &= \|\bar{A}^T P + P\bar{A}\|_2 \\ &\leq 2\|P\bar{A}\|_2 \\ &\Rightarrow \frac{1}{2}\|\bar{C}^T\bar{C}\|_2 \leq \|P\bar{A}\|_2 \\ &\leq \|P\|_2\|\bar{A}\|_2 \\ &= \overline{\sigma}(P)\|\bar{A}\|_2. \end{aligned}$$

Note that $\|\bar{A}\|_2 = \overline{\sigma}(A) = \sqrt{\bar{\lambda}(\bar{A}^T\bar{A})}$ and $\bar{A}^T\bar{A}$ can be factored as follows

$$\begin{aligned} \bar{A}^T\bar{A} &= \begin{bmatrix} A_{11}^T A_{11} + A_{21}^T A_{21} & A_{11}^T A_{12} + A_{21}^T A_{22} & 0 \\ A_{12}^T A_{11} + A_{22}^T A_{21} & A_{12}^T A_{12} + A_{22}^T A_{22} & 0 \\ 0 & 0 & A_{33}^T A_{33} \end{bmatrix} + \widetilde{B} \\ &= \operatorname{diag}(A^T A, A_{33}^T A_{33}) + \widetilde{B} \\ &\triangleq \widetilde{A} + \widetilde{B}. \end{aligned}$$

Clearly $\widetilde{A} + \widetilde{B}$ is symmetric, thus following from inequality (6.37a) we have

$$\begin{aligned} \frac{1}{2}\|\bar{C}^T\bar{C}\|_2 &\leq \overline{\sigma}(P)\left[\overline{\sigma}(\widetilde{A} + \widetilde{B})\right] \\ &\leq \overline{\sigma}(P)\left[\overline{\sigma}(\widetilde{A}) + \overline{\sigma}(\widetilde{B})\right] \\ &= \overline{\sigma}(P)\left[\max\{\overline{\sigma}(A), A_{33}\} + \overline{\sigma}(\widetilde{B})\right] \\ &\Rightarrow \frac{1}{2}\|\bar{C}^T\bar{C}\|_2\left[\max\{\overline{\sigma}(A), A_{33}\} + \overline{\sigma}(\widetilde{B})\right]^{-1} \leq \overline{\sigma}(P) \end{aligned}$$

noting that $P = P^T$ and therefore $\overline{\sigma}(P) = \bar{\lambda}(P)$ completes the proof. ■

Remark 7 *Also observe that* $\frac{1}{2}\|\bar{C}^T\bar{C}\|_2 = \frac{1}{2}max\{\overline{\sigma}(C), \overline{\sigma}(C_{33})\}$.

This theorem provides a lower bound on the output energy of an appended

system in a decentralized manner, using just local information about the two subsystems. It is straight forward to generalize Theorem 12 to the case where A_{33} is any Hurwitz matrix and C_{33} is any matrix of appropriate dimension. We have also implicitly assumed that there are no external inputs to the state being appended (or at least chosen to ignore them). Such a decision balances the accuracy of the estimate with computational efficiency. In Section 6.5.1.4 this assumption is removed.

Corollary 2 *A lower bound of the largest singular value of the matrix $X \in \mathcal{S}_{++}$ that solves the Lyapunov equation $A^TX + XA + Q = 0$ with $Q = C^TC$ is given by*

$$\overline{\sigma}(X) \geq \frac{\overline{\sigma}(Q)}{2\overline{\sigma}(A)}.$$

Proof: The inequality follows immediately by setting $A_{33} = B_1 = B_2 = 0$ in Theorem 12. ■

The following theorem is dual to the one just presented where we now consider appending an additional state (or system) *up-stream* of the existing system corresponding to Figure 6.4 (Middle). The proof follows closely the arguments of Theorem 12 and so is ommited for brevity.

Theorem 13 *Assume that $B_1 = 0$ and $B_2 = 0$ in (6.34) and that the matrices A and A_{33} are Hurwitz. The maximum $\mathcal{L}_2$ norm of the output satisfies the inequality*

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \;\geq\; \frac{1}{2}\|\bar{C}^T\bar{C}\|_2 \cdot \left[max\{\overline{\sigma}(A), \overline{\sigma}(H)\} + \overline{\sigma}(\widehat{B})\right]^{-1}$$

where $H = B_3^TB_3 + B_4^TB_4 + A_{33}^TA_{33}$ and

$$\widehat{B} = \begin{bmatrix} 0 & 0 & A_{11}^TB_3 + A_{21}^TB_4 \\ 0 & 0 & A_{12}^TB_3 + A_{22}^TB_4 \\ * & * & 0 \end{bmatrix}.$$

Unsurprisingly, unlike in Theorem 12, the up-stream result in Theorem 13 does

not decouple perfectly. However, the coupling is minimal (only present in $\widehat{B}$) and the structure of $\widehat{B}$ is convenient from a numerical perspective, i.e. it is symmetric, sparse and block partitioned. In this case we can consider the results to be distributed rather than decentralized.

Finally for the case where the appended state interacts with both up and down-stream states, i.e. $B_i \neq 0$ for $i = 1, \ldots, 4$ the inequality in Theorem 13 should be changed to

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \geq \frac{1}{2}\|\bar{C}^T\bar{C}\|_2 \left[\max\{\overline{\sigma}(A), \overline{\sigma}(H)\} + \overline{\sigma}(\widehat{B} + \widetilde{B})\right]^{-1}. \tag{6.38}$$

Further decoupling is possible at the expense of accuracy of the attainable bound. The term $\overline{\sigma}(\widehat{B} + \widetilde{B})$ in (6.38) can be replaced by $\overline{\sigma}(\widehat{B}) + \overline{\sigma}(\widetilde{B})$ thus we now have the same amount of coupling as in Theorem 13.

6.5.1.2 Upper bound estimates

In this section we obtain decentralized upper bound estimates of the $\mathcal{L}_2$ energy of a system when a state is appended. This information can, in addition to the lower bound estimates, be incorporated into a clustering based decomposition algorithm as will be demonstrated in the following section. We make use of a result taken from [121]:

Lemma 5 *An upper bound for the largest singular value of the matrix $X \in \mathcal{S}_{++}$ that satisfies $A^TX + XA + Q = 0$ where Q is full rank and given by $Q = C^TC$ is:*

$$\overline{\sigma}(X) \leq -\frac{1}{2\overline{\lambda}(SQ^{-1})} \tag{6.39}$$

where $S = (A + A^T)/2$.

What is not immediately obvious from Lemma 5 is that the eigenvalues of SQ^{-1} are always real and negative despite the fact that SQ^{-1} is not symmetric. This fact follows from a result of *congruent matrices* and an application of *Sylvester's*

law of inertia [30]. For completeness the result is given in Appendix A.3 of this chapter.

In order to simplify notation later let us define the operator $\Psi : \mathcal{R}^{k\times k} \to \mathcal{R}^{k\times k}$ by $\Psi(B) = (B + B^T)/2$ where $B \in \mathcal{R}^{k\times k}$. We will also assume that the output matrices C and C_{33} are identity as we do not require output maps in the forthcoming clustering algorithm. In order to compute estimates for $\overline{\sigma}(X)$ in a distributed manner as was done for the lower bound estimates it is desirable to replace the eigenvalue-based conditions from Lemma 5 with singular value bounds.

Under the assumption that the output maps are identity matrices the denominator in (6.39) can be equivalently expressed in terms of singular values using the following lemma.

Lemma 6 *Given the matrix $F \prec 0$ then $\overline{\lambda}(F) = -\underline{\sigma}(F)$.*

Applying the above Lemma, setting $Q = I$ and using the fact that $S = S^T$ the inequality in Lemma 5 becomes:

$$\overline{\sigma}(X) \leq \frac{1}{2\underline{\sigma}(S)}. \tag{6.40}$$

Before stating the main result of this section we require one further result known as Fan's theorem. We present a simplified version as presented in [122].

Lemma 7 *Let $Y, Z \in \mathcal{R}^{n\times m}$ be given and let $q = min\{m, n\}$. Assuming that the singular values of Y and Z are in decreasing order then*

$$\sigma_{i+j-1}(Y + Z) \leq \sigma_i(Y) + \sigma_j(Z).$$

In particular when $i = 1$, $j = n$ and $m = n$:

$$\underline{\sigma}(Y + Z) \leq \underline{\sigma}(Y) + \overline{\sigma}(Z).$$

Based on the results above distributed upper bounds for the maximum $\mathcal{L}_2$ gain of the output signal of a system are now presented.

Theorem 14 *Given System (6.34) with $B_3 = B_4 = 0$ and $C = I$, $C_{33} = 1$, if A and A_{33} are Hurwitz matrices then*

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \leq \frac{1}{2(min\ \{\underline{\sigma}[\Psi(A)], \underline{\sigma}[\Psi(A_{33})]\} + \overline{\sigma}[\Pi])}$$

where

$$\Pi = \begin{bmatrix} 0 & \frac{1}{2}B_{(12)}^T \\ \frac{1}{2}B_{(12)} & 0 \end{bmatrix}$$

with $B_{(12)} = [B_1\ B_2]$.

Proof: As we have shown already, the left hand side of the inequality in the theorem is given by the maximum singular value of the positive definite matrix that solves the Lyapunov equation. What remains to be shown is that the right hand side is a distributed representation of the right hand side of the inequality in Lemma 5. We have that

$$\Psi(\bar{A}) = \frac{1}{2}\begin{bmatrix} A_{11} + A_{11}^T & A_{12} + A_{21}^T & B_1^T \\ A_{21} + A_{12}^T & A_{22} + A_{22}^T & B_2^T \\ B_1 & B_2 & A_{33} + A_{33}^T \end{bmatrix} = \begin{bmatrix} \Psi(A) & \frac{1}{2}B_{(12)}^T \\ \frac{1}{2}B_{(12)} & \Psi(A_{33}) \end{bmatrix}.$$

By assumption we have that $Q = \bar{C}^T\bar{C} = I$ and so $Q^{-1} = Q = I$ and therefore $\Psi(\bar{A})Q^{-1} = \Psi(\bar{A})$ which allows us to use the bound given by (6.40). A further factorization of $\Psi(\bar{A})$ gives:

$$\underline{\sigma}[\Psi(\bar{A})] = \underline{\sigma}\left(\begin{bmatrix} \Psi(A) & 0 \\ 0 & \Psi(A_{33}) \end{bmatrix} + \Pi\right)$$

which by application of Lemma 7 is upper bounded by

$$\min\{\underline{\sigma}[\Psi(A)], \underline{\sigma}[\Psi(A_{33})]\} + \overline{\sigma}[\Pi]$$

completing the proof. ■

In comparison to the lower bound estimates it can be seen that the coupling enters in a "cleaner" additive manner through the operator Ψ. The addition of states up stream and in feedback follow in exactly the same manner:

Corollary 3 *Given System (6.34) where A and A_{33} are Hurwitz matrices and $C = I$ and $C_{33} = 1$ then:*

1. *When $B_1 = B_2 = 0$*

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \leq \frac{1}{2(min\{\underline{\sigma}[\Psi(A)], \underline{\sigma}[\Psi(A_{33})]\} + \overline{\sigma}[\widehat{\Pi}])}$$

where

$$\widehat{\Pi} = \begin{bmatrix} 0 & \frac{1}{2}B_{(34)}^T \\ \frac{1}{2}B_{(34)} & 0 \end{bmatrix}$$

with $B_{(34)} = [B_3^T \ B_4^T]^T$.

2. *When $B_1, B_2, B_3, B_4 \neq 0$*

$$\max_{\|x_0\|_2^2=1} \|y\|_{\mathcal{L}_2}^2 \leq \frac{1}{2(min\{\underline{\sigma}[\Psi(A)], \underline{\sigma}[\Psi(A_{33})]\} + \overline{\sigma}[\widetilde{\Pi}])}$$

where

$$\widetilde{\Pi} = \begin{bmatrix} 0 & 0 & B_3 + B_1^T \\ 0 & 0 & B_4 + B_2^T \\ B_1 + B_3^T & B_2 + B_4^T & 0 \end{bmatrix}.$$

The results above and in Theorem 14 focussed on the case where both C and C_{33} where identity matrices of appropriate dimension. We will now outline why

the case for general output maps is problematic when considering distributed computation.

From Lemma 5 we have that $\overline{\sigma}(X) = \max\|y\|_{\mathcal{L}_2}^2 \leq [-2\overline{\lambda}(SQ^{-1})]^{-1}$. The main problem preventing the development of a distributed method for computing an upper bound on $\overline{\sigma}(X)$ is that the matrix SQ^{-1} is not symmetric. Using the notation introduced above we have that $Q^{-1} = \text{diag}((C^TC)^{-1}, (C_{33}^TC_{33})^{-1})$ and $S = \Psi(\bar{A})$ as given in Theorem 14. Note that this implicitly requires that C^TC and $C_{33}^TC_{33}$ have full rank. The inequality of Lemma 5 is now

$$\overline{\sigma}(X) \leq -\frac{1}{2\overline{\lambda}(\Delta)},$$

where

$$\Delta = \begin{bmatrix} \Psi(A)(C^TC)^{-1} & \frac{1}{2}B_{12}^T(C_{33}^TC_{33})^{-1} \\ \frac{1}{2}B_{12}(C^TC)^{-1} & \Psi(A_{33})(C_{33}^TC_{33})^{-1} \end{bmatrix}.$$

At this point in the previous analysis the denominator was expressed in terms of singular values, the fact that the bound above is written in terms of eigenvalues makes the factorization more complicated. A possible solution to this problem is to split Δ into a block diagonal matrix and off-diagonal matrix and use a block matrix version of the Geršgorin disk theorem [93]. Such an approach will be the focus of future research.

6.5.1.3 Clustering Algorithm

Using the theory developed in the previous section a clustering based algorithm is described that *lumps* states together based on the estimated energy (we use the term energy to mean the maximum achievable $\mathcal{L}_2$ norm of a signal) observed at the output. Based on the work of Chapter 4, states are clustered together if their output energy estimate, i.e. $\|y\|_{\mathcal{L}_2}^2$ is small in comparison to other possible clusters. We make the same assumption that there is no coupling between the output maps of the clustered subsystems. The algorithm described is general and

captures the salient features of a clustering algorithm (several variants on the algorithm are possible):

1. Represent system $\dot{x} = Ax$, $y = Cx$ as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where the graph structure is defined by the adjacency matrix $A(\mathcal{G})$ given by (4.21).

2. Assign weights to the edges of the graph based on the estimated energy ($\|y\|^2_{\mathcal{L}_2}$) of the lumped system. Note that this weight depends on whether the edge is bidirectional (in which case the feedback estimate is used) or unidirectional (in which case use the up/down stream estimates). At this stage we now have a directed weighted graph $G(\mathcal{V}, \mathcal{E}, \mathcal{Z})$.

3. Lump together the nodes which have the smallest edge weight connection.

4. Update the graph taking into account the new lumped node(s) and go to step 2.

When the clustering algorithm terminates, the resulting system decomposition can then be tested for stability using composite methods such as those described previously. In the following section a small-gain type stability proof will be produced. Note that Step 3 in the algorithm above allows for several possibilities. It may be desirable to simply lump together two states and then update the system graph and repeat the process. Alternatively one could lump k states together before updating the network. Another variation of the algorithm concerns the termination criterion. It is possible to stop the algorithm once there is an equal division of states amongst clusters, or the algorithm can be left to end naturally with multiple subsystems of varying dimension, or simply if the energy flow between remaining subsystems is deemed as too large. The following example illustrates one variant of the algorithm.

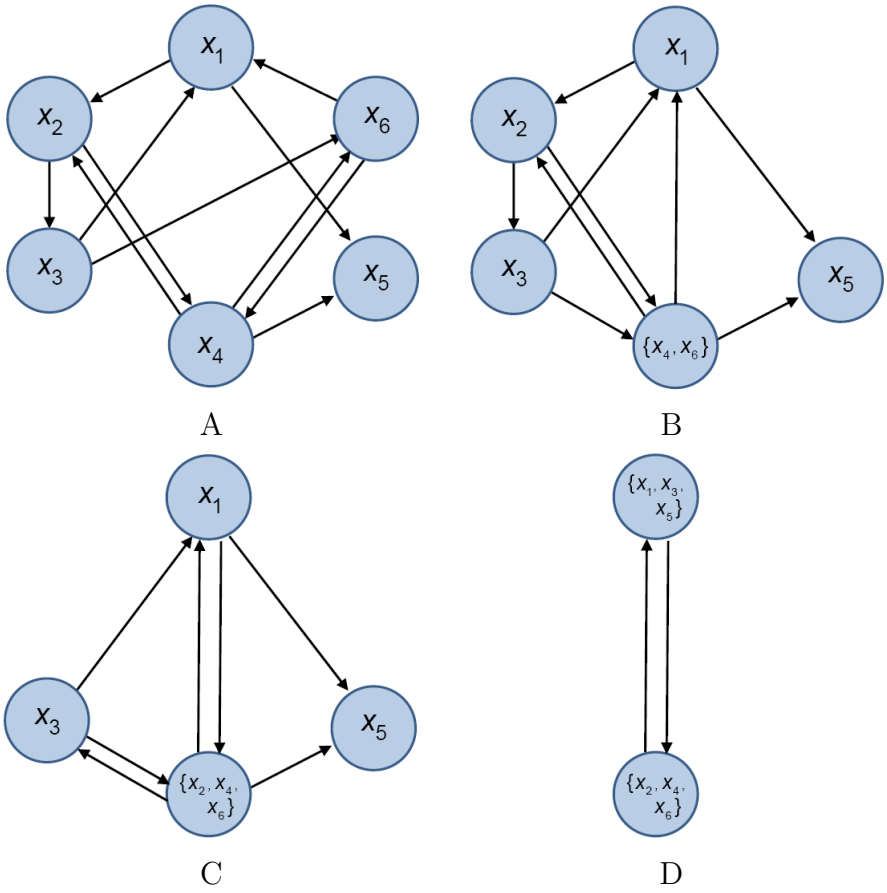


Figure 6.5: Step-by-step clustering algorithm example. System (6.41) is first represented as a graph (A). Iteration 1 (B), 2 (C) and (3) of the algorithm.

Consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -0.261 & 0 & 1.021 & 0 & 0 & -0.001 \\ 0.357 & -1.495 & 0 & 0.564 & 0 & 0 \\ 0 & -2.301 & -1.138 & 0 & 0 & 0 \\ 0 & -0.698 & 0 & -0.593 & 0 & -0.768 \\ 0.015 & 0 & 0 & 0.456 & -0.639 & 0 \\ 0 & 0 & 2.438 & 1.132 & 0 & -1.046 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{6.41}$$

with the output $y = x$. The graph of this system is shown in Figure 6.5 (A). The second step of the algorithm assigns weights to the edges of the graph. In this example we must compute energy estimates for the 9 possible clusters (corresponds to 9 edges in the graph), illustrated in Figure (6.6). The energy

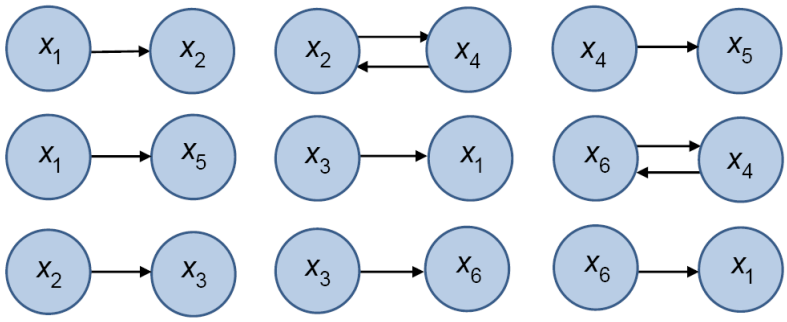


Figure 6.6: Possible clusters at the first iteration of the clustering algorithm.

estimates of the clusters are assigned to the edge weights. It is found that the smallest weight corresponds to the feedback connection of x_4 and x_6. These states are clustered together and the network is rewired appropriately as shown in Figure 6.5(B). In the next iteration we must compute gain estimates for clusters that involve the new lumped node, here this corresponds to a feedback connection with x_2 and up/down-stream connections with x_1, x_3, x_5. As the algorithm proceeds x_2 is clustered with x_6 and x_4, Figure 6.5(C), at which point we cluster the remaining states in order to have two subsystems with an equal number of states, Figure 6.5(D). A composite Lyapunov function for the decomposition is found thus verifying the stability of (6.41).

6.5.1.4 Input-Output Clustering

In the previous section the bounds we obtained did not take into account any input dynamics on the nodes. In this section we explicitly take into account the input and outputs and consider the $\mathcal{L}_2 \to \mathcal{L}_2$ gains of the nodes. The trade off is that this approach incurs a greater computational cost as an SDP must be solved for each node instead of a Lyapunov inequality.

To obtain upper bound estimates of the system gain we will use the fact that the $\mathcal{L}_2 \to \mathcal{L}_2$ gain (6.14) of an LTI system (6.5) is the $\mathcal{H}_\infty$ norm as defined earlier. We will exploit the fact that the $\mathcal{H}_\infty$ norm posseses a sub-multiplicative property,

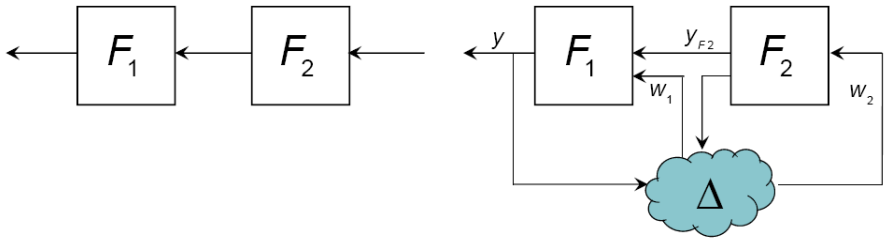


Figure 6.7: Serial connection of two LTI systems, with (right) and without (left) external disturbance. The external disturbance is caused by states which feed into F_1 or F_2 that do not belong to either system. The Δ term represents the *rest of the network*.

i.e. if F_1 and F_2 are two LTI systems in a cascade as shown in Figure 6.7 then

$$\|F_1F_2\|_{\mathcal{H}_\infty} \leq \|F_1\|_{\mathcal{H}_\infty} \cdot \|F_2\|_{\mathcal{H}_\infty}. \tag{6.42}$$

The result above is easily seen by noting that $\|F_2F_1\|_{\mathcal{H}_\infty} = \sup_{\omega\in\mathcal{R}}|F_2(j\omega)F_1(j\omega)|$ while $\|F_2\|_{\mathcal{H}_\infty} \cdot \|F_1\|_{\mathcal{H}_\infty} = (\sup_{\omega\in\mathcal{R}}|F_2(j\omega)|)(\sup_{\omega\in\mathcal{R}}|F_1(j\omega)|)$. Note also that in general $\|F_1F_2\|_{\mathcal{H}_\infty} \neq \|F_2F_1\|_{\mathcal{H}_\infty}$.

Let us first consider the down-stream case, we will focus on appending a single state system F_1 downstream of an existing two state system F_2. We must take into account the fact that the input to F_1 consists of the output of F_2 which is denoted by y_{F2} and inputs from states which are not part of F_2 which will be denoted by the vector w_1; this is illustrated in Figure 6.7 (Right).

Let F_2 be the existing system which has the state space realization

$$F_2 \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} w_2 \\ \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases}$$

where H_{11} and H_{22} are row vectors (of appropriate dimension) and w_2 is the

vector of external states whose input feeds into F_2; together these define an input map. The $\mathcal{L}_2 \to \mathcal{L}_2$ ($\mathcal{H}_\infty$) norm of this type of system can be computed exactly via semidefinite programming using the KYP-Lemma [5,17]. The associated SDP and a version of the KYP-Lemma are given in Appendix A.2 of this chapter.

We consider the case where the output of F_2, y_{F2}, is connected directly to the input of F_1. The state space representation of F_1 is then given by

$$F_1 \begin{cases} \dot{x}_3 = A_{33}x_3 + [B_1 \ B_2]y_{F2} + Hw_1 \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_3 + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} y_{F2} \end{cases}.$$

To compute $\|F_1\|_{\mathcal{H}_\infty}$ exactly we must solve the SDP corresponding to the *KYP Lemma* which in this case takes the form:

$$\begin{aligned} \min_{P_3 \in \mathcal{S}, \gamma_1} \quad & \gamma_1 \\ \text{s.t.} \quad & P_3 \succ 0, \quad \gamma_1 > 0 \\ & \begin{bmatrix} A_{33}^T P_3 + P_3 A_{33} + 1 & P_3 \begin{bmatrix} B_1 & B_2 & H \end{bmatrix} \\ \begin{bmatrix} B_1^T \\ B_2^T \\ H^T \end{bmatrix} P_3 & I - \gamma_1 \end{bmatrix} \preceq 0 \end{aligned} \tag{6.43}$$

It then follows that $\|F_1\|_{\mathcal{H}_\infty} = \sqrt{\gamma_1}$.

Using the two methods presented an upper bound for the $\mathcal{L}_2$ norm of the output of a system when a state is appended down-stream is given by

$$\sqrt{\gamma_1}\sqrt{\gamma_2} \geq \|F_1 F_2\|_{\mathcal{H}_\infty}$$

where $\sqrt{\gamma_2} = \|F_2\|_{\mathcal{H}_\infty}$.

Let us now consider the case of adding a single state up-stream of an exist-

ing subsystem. The argument proceeds in the same manner as above with the exception that F_2 is now defined as

$$F_2 \begin{cases} \dot{x}_3 = A_{33}x_3 + Hw_2 \\ y_3 = x_3 \end{cases}$$

where we can compute $\|F_2\|_{\mathcal{H}_\infty}$ using the KYP-lemma in the usual manner. The down-stream system F_1 is now given by

$$F_1 \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_3 \\ B_4 \end{bmatrix} y_3 + \begin{bmatrix} H_3 \\ H_4 \end{bmatrix} w_1 \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} y_3 \end{cases}$$

Then the SDP that computes $\|F_1\|_{\mathcal{H}_\infty}$ is:

$$\begin{aligned} \min_{P \in \mathcal{S}, \gamma_1} \quad & \gamma_1 \\ \text{s.t.} \quad & P \succ 0, \quad \gamma > 0 \\ & \begin{bmatrix} A^TP + PA + C^TC & P \begin{bmatrix} B_3 & H_3 \\ B_4 & H_4 \end{bmatrix} \\ \begin{bmatrix} B_3^T & B_4^T \\ H_3^T & H_4^T \end{bmatrix} P & \begin{bmatrix} 1-\gamma_1 & 0 \\ 0 & -\gamma_1 \end{bmatrix} \end{bmatrix} \preceq 0 \end{aligned} \tag{6.44}$$

and thus $\|F_1\|_{\mathcal{H}_\infty} = \sqrt{\gamma_1}$ from which we immediately have the upper-bound on the $\mathcal{L}_2 \to \mathcal{L}_2$ given by $\sqrt{\gamma_1}\sqrt{\gamma_2}$.

Finally we deal with the case where nodes are connected in positive feedback. In this instance we have to arrange our subsystems such that they take the form given in Figure 6.8 in order to be able to apply the small-gain theorem and obtain

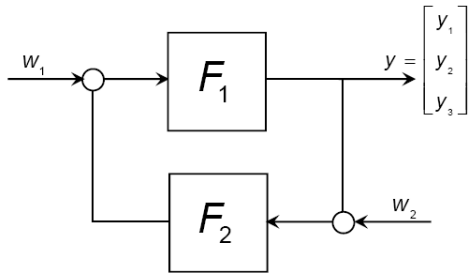


Figure 6.8: Feedback connection

a bound on the gain. With this subsystem arrangement F_1 and F_2 are given by:

$$F_2 \begin{cases} \dot{x}_1 = A_{33}x_3 + [B_1 \quad B_2]\, y_{F2} + Hw_2 \\ y_3 = x_3 \end{cases}$$

and

$$F_1 \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_3 \\ B_4 \end{bmatrix} y_3 + \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} w_1 \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} y_3 \end{cases}$$

First we apply the KYP lemma and solve the relevant SDPs to obtain the system norms $\|F_1\|_{\mathcal{H}_\infty}$ and $\|F_2\|_{\mathcal{H}_\infty}$. A direct application of the small-gain theorem produces the following bound on the system output:

$$\|F_1 \circ F_2\|_{\mathcal{H}_\infty} \leq \frac{\|F_1\|_{\mathcal{H}_\infty}}{1 - \|F_1\|_{\mathcal{H}_\infty}\|F_2\|_{\mathcal{H}_\infty}} \tag{6.45}$$

where $\circ$ denotes the positive feedback connection.

When considering cascade interconnections there are two degenerate cases that may arise. These correspond to when the upstream subsystems do not have any inputs and when downstream subsystems do not have any outputs. When this

happens there are two possible options: i) Lump the node into a neigbouring node (essentially this ensures that the node is ignored by the clustering algorithm). ii) Compute the maximum $\mathcal{L}_2$ norm of the output[1] signal by solving a Lyapunov equation and use this in place of the upstream system's $\mathcal{H}_\infty$ norm in the bound estimate calculations.

The results described in this section can be incorporated into a clustering algorithm in exactly the same manner as in the previous section. It is acknowledged however that the computational cost of computing the input-output bounds is larger than that of computing the energy bounds. However some computational headway can be made by incorporating a *small gain* heuristic into the algorithm. Such a heuristic would simply lump together any nodes automatically if the product of the $\mathcal{H}_\infty$ norms two systems in feedback is less than one.

The following example demonstrates an input-output clustering algorithm: Consider the stable system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -1.658 & 3.383 & 0 & 0 & 2.97 & -1.169 \\ 0 & -1.735 & 0.717 & -2.859 & -0.522 & 0 \\ -0.093 & -0.851 & -2.078 & -1.442 & 0 & 0 \\ 3.749 & 0 & 1.077 & -1.867 & 0 & 0 \\ 0 & 0.991 & 1.430 & 0 & -2.079 & -0.804 \\ 0 & 0 & 0 & 0 & 0.347 & -1.957 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

First the system is represented as a graph in accordance with Step 1 of the clustering algorithm. In the first instance each node represents a multi-input, single output LTI system. We begin by applying the small gain heuristic, this involves computing the $\mathcal{H}_\infty$ norm for each system that is connected in positive feedback. It is found that nodes 3 and 4, and nodes 5 and 6 satisfy the small gain condition and thus we can immediately cluster them together. After redrawing the graph to take into account the two new subsystems we are left with a four node directed

[1] This is only applicable to upstream subsystems.

graph. At Step 2 of the clustering algorithm edge weights are assigned. Here edge weights ($\mathcal{L}_2 \to \mathcal{L}_2$ gains) are computed based the type of interconnection (feedback, up/down stream interconnection). Once all weights are computed nodes connected by the smallest weighted edge are clustered together. In this example the smallest edge weight corresponds to connecting node 1 with the nodes 5 and 6, followed by node 2 with nodes 3 and 4. The resulting decomposed subsystems are

$$\mathcal{T}_1 = \{x_1, x_5, x_6\}\,, \quad \mathcal{T}_2 = \{x_2, x_3, x_4\}\,.$$

Both subsystems are shown to asymptotically stable when the input is set to zero, the feedback interconnection of the subsystems is also shown to be stable.

Top down or bottom up?

We have now presented three algorithms for system decomposition. In Chapter 4 a top down method was presented while in this chapter a bottom up approach based on clustering was described that uses either upper or lower bound estimates as described in Section 6.5.1.1 and Section 6.5.1.2 respectively. Finally in Section 6.5.1.4 a clustering algorithm that specifically takes into account the subsystem interconnections was presented.

At the heart of all the algorithms presented there was a unifying theme that minimizing the energy flow between subsystems would enhance the likelihood of being able to verify the stability of the original large-scale system. The mechanisms for producing the decompositions are graph partitioning (top down) or graph clustering (bottom up). Comparing the three methods is difficult as they use different stability tests: The partitioning approach relies on the existence of composite Lyapunov functions where the interconnection dynamics are ignored. In contrast the bottom up approach uses small-gain type stability proofs. The two approaches compliment each other well. In particular it was shown that if

the two subsystems satisfy the small gain condition (a bottom up method) then a block diagonal composite Lyapunov function must exist, the tradeoff here is that the local analysis of computing the system gain is computationally more expensive than the top down approach. In comparison, the input output clustering algorithm carries the largest computational burden as an SDP must be solved for every edge in the graph, this process is then repeated again after nodes/states have been lumped together.

A factor to consider at this point is that the top down method uses a graph partitioning algorithm. Both spectral and Positivstellensatz based algorithms that we described attempt to find the optimal partition in a one shot approach. In contrast the clustering algorithms are more likely to get stuck in a local minimum (possibly resulting is suboptimal decompositions and thus fail to verify stability) as they rely on greedy (node by node) algorithms. Thus if a clustering based approach fails there is no reason to assume that a top down approach will also fail.

6.6 Further Remarks on Tractability

As was done at the end of Chapter 4 in Section 4.5 we conclude with remarks on why a direct approach to solving the clustering based decomposition problem is theoretically and computationally very difficult and hence the need for good heuristic solutions. A direct solution involves solving an optimization problem over the group of permutation matrices Π_n. The remainder of this chapter shows that solving the clustering based decomposition problem can be formulated as a modified Quadratic Assignment Problem (QAP) - a known $\mathcal{NP}$-hard problem.

6.6.1 Optimization over Π

The outline of a convex optimization based relaxation algorithm for graph clustering is presented here. The main technique used is a convex relaxation to

the Quadratic Assignment Problem (QAP). First the proposed relaxation to the decomposition-clustering algorithm is presented and then an outline of the derivation of the QAP relaxation that we rely on is presented for completeness.

A Convex Relaxation of the Decomposition-Clustering Algorithm

Assume that the system to be decomposed, G is given by $\dot{x} = Fx$ with $x \in \mathcal{R}^n$ and that the objective is to decompose G into two smaller subsystems G_1, G_2 that satisfy the small-gain theorem.

Let A be a diagonal $n \times n$ matrix such that $A_{ii} = \gamma_i$ where γ_i is the $\mathcal{H}_\infty$ norm of the single state subsystem

$$\begin{aligned} \dot{x}_i &= F_{ii}x_i + F_i^{'}x^{'} \\ y_i &= x_i \end{aligned}$$

where F_i' is the i^{th} row of F excluding the F_{ii}, and x' is the matching state vector. This corresponds to the node description from Section 6.5.1.4. The matrix A is found by solving the SDP associated to the KYP lemma n times. Let $\tilde{A}$ denote $\log(A)$ and define

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \in \mathcal{S}^n, \quad \tilde{P}_1 = \begin{bmatrix} P_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{P}_2 = \begin{bmatrix} 0 & 0 \\ 0 & P_2 \end{bmatrix}.$$

Assume $P = I$ and G has been decomposed into 2 subsystems G_1 and G_2 with states $x^{(1)} = [x_1, \ldots, x_m]^T$ and $x^{(2)} = [x_{m+1}, \ldots, x_n]^T$. If F was a full matrix, i.e. every node in the graph is connected to every other node, and we have computed γ_i for each node then the product of these gains for one of the subsystems is given by

$$\gamma_i^* = e^{\text{Tr}(\tilde{P}_i \tilde{A} \tilde{P}_i)}. \tag{6.46}$$

For different system decompositions resulting from alternative state permutations

(6.46) is replaced by

$$\gamma_i^* = e^{\mathrm{Tr}(\tilde{P}_i X \tilde{A} \tilde{X}^T P_i)}, \tag{6.47}$$

where $X \in \Pi$. Recall that $e^{\alpha x}$ is convex on $\mathcal{R}$ for any $\alpha \in \mathcal{R}$ [6]. The optimization problem we now wish to solve is given by

$$\begin{aligned} \min_{X} \quad & \mathrm{Tr}(E) \\ \text{s.t.} \quad & XX^T = X^T X = I \\ & \|X\mathbf{1} - \mathbf{1}\|_2^2 + \|X^T\mathbf{1} - \mathbf{1}\|_2^2 = 0 \\ & X_{ij}^2 - X_{ij} = 0, \quad \forall i, j \\ & e^{\mathrm{Tr}(\tilde{P}_1 X \tilde{A} \tilde{X}^T P_1)} \leq \epsilon_1 \\ & e^{\mathrm{Tr}(\tilde{P}_2 X \tilde{A} \tilde{X}^T P_2)} \leq \epsilon_2 \end{aligned} \tag{6.48}$$

where $E = \begin{bmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{bmatrix}$. Using the QAP results to be presented in the next section it will be shown that it is actually the Lagrangian dual [6] of (6.48) that we should aim to solve.

The optimization (6.48) corresponds to finding subsystems that have a small loop gain in addition to satisfying the small gain theorem if ϵ_i $i = 1, 2$ can be made sufficiently small such that $\epsilon_1 \epsilon_2 < 1$. It is unlikely that this condition will be satisfied directly as we are using combinations of upper bounds. Ideally, once the optimization has been solved the associated subsystems are then constructed and the appropriate SDPs are solved to determine if the finite gain theorem has been satisfied.

We stress that this is future work as there are likely many technical details that need to be addressed. The method does however have the potential to help relieve part of the combinatorial issues related to system decomposition although this is at the expense of computational efficiency. In the following section a convex

relaxation of the QAP is presented that explains where the structure of (6.48) comes from.

The Quadratic Assignment Problem

The QAP is a well studied non-convex optimization problem over the manifold of permutation matrices Π. The standard formulation is:

$$\mu^* = \underset{X \in \Pi}{\arg\min} \quad \mathrm{Tr}(AXBX^T - 2CX^T) \tag{6.49}$$

where $A, B \in \mathcal{S}^n$ and $C \in \mathcal{R}^{n \times n}$ are given matrices. QAP is known to be $\mathcal{NP}$-hard, furthermore, problems where $n \geq 16$ are considered large-scale. We begin by outlining how the dual of the Lagrangian dual is used to obtain a convex problem. We follow the solution methodology and notation of [123] and [124]. Begin by considering a simplified version of QAP:

$$\begin{aligned} \delta^* = \underset{X}{\arg\min} \quad & \mathrm{Tr}(AXBX^T) \\ \text{s.t.} \quad & XX^T = I \end{aligned} \tag{6.50}$$

Optimization (6.50) is minimizing a quadratic function subject to the decision variable X being an orthogonal matrix. This problem can be solved exactly using standard Lagrange multiplier methods. An outline of the proof is given here: The Lagrangian of (6.50) is $L(X, S) = \mathrm{Tr}(AXBX^T) - \mathrm{Tr}S(XX^T - I)$. It is shown that the neccessary and sufficient conditions for optimality are that the gradient of the Lagrangian is zero, i.e.

$$\begin{aligned} \frac{\partial L(X,S)}{\partial X} &= \frac{\partial \mathrm{Tr}(AXBX^T) - \mathrm{Tr}S(XX^T - I)}{\partial X} = 0, \\ &\Rightarrow \quad AXB - SXI = 0, \\ &\Rightarrow \quad AXBX^T = S. \end{aligned}$$

As the term $AXBX^T$ is symmetric it follows that $S = S^T$. The optimal solution of (6.50) is

$$\delta^* = \alpha^T \beta, \tag{6.51}$$

where α is the vector of eigenvalues of A in nonincreasing order and β is the vector of eigenvalues of B ordered in a nondecreasing manner. The orthogonal matrix X is then recovered by setting $X = VU^T$ where V and U the orthonormal diagonalizing basis vectors for A and B respectively.

The Lagrangian dual of (6.50) is

$$\max_{S=S^T} \quad \min_{X} \quad \mathrm{Tr}(AXBX^T) - \mathrm{Tr}S(XX^T - I). \tag{6.52}$$

The inner optimization problem in (6.52) is an unconstrained quadratic minimization, thus it is possible that a non-zero duality gap exists. Let $\mathbf{x} = \mathrm{vec}(X)$ and $\otimes$ denote the Kronecker product, then the Hessian of the minimization problem is

$$\mathbf{x}^T(B \otimes A - I \otimes S)\mathbf{x}$$

which is seen by observing that $\mathrm{Tr}(AXBX^T) = \mathbf{x}^T(B \otimes A)\mathbf{x}$. If $(B \otimes A - I \otimes S) \succeq 0$ does not hold then the minimization is unbounded. This problem is overcome by adding a seemingly redundant constraint; Orthogonal matrices satisfy the relation $XX^T = I = X^T X$. Including this additional (but equivalent) constraint gives

$$\begin{aligned} \delta^* = \arg\min_{X} \quad & \mathrm{Tr}(AXBX^T) \\ \text{s.t.} \quad & XX^T = I, \quad X^T X = I \end{aligned} \tag{6.53}$$

and its Lagrangian dual:

$$\begin{aligned} \max_{S=S^T, T=T^T} \quad & \mathrm{Tr}(S) + \mathrm{Tr}(T) \\ \text{s.t.} \quad & (I \otimes S) + (T \otimes I) \preceq (B \otimes A) \end{aligned} \tag{6.54}$$

where S and T are the Lagrangian dual variables for the constraints $XX^T = I$ and $X^TX = I$ respectively. The additional constraint corresponds to the Hessian matrix which must satisfy a positivity requirement in order for the optimization to remain bounded. Strong duality between (6.53) and (6.54) was proven in [123]. The SDP (6.54) will obtain exactly the optimal value (6.51).

The problem we wish to solve is more complicated, we want X to be a permutation matrix rather than simply orthogonal. Let $\mathcal{E}$ denote the set of square metrices with row and column sums equal to one, $\mathcal{O}$ denotes the group of square orthogonal matrices and $\mathcal{N}$ the set of square matrices with nonnegative real entries:

$$\begin{aligned}\mathcal{E} &= \{X \in \mathcal{R}^{n\times n} \mid X\mathbf{1} = X^T\mathbf{1} = 1\},\\ \mathcal{O} &= \{X \in \mathcal{R}^{n\times n} \mid XX^T = X^TX = I\},\\ \mathcal{N} &= \{X \in \mathcal{R}^{n\times n} \mid X \geq 0\}.\end{aligned}$$

The set of doubly stochastic matrices is then given by $\Omega = \mathcal{E} \cap \mathcal{N}$. It then follows that the group of permutation matrices Π, satisfies $\Pi = \mathcal{O} \cap \Omega = \mathcal{O} \cap \mathcal{E} \cap \mathcal{N}$. Additionally, $\Pi = \mathcal{O} \cap \mathcal{N}$. Thus it can be seen that the optimization problem described previously goes some way towards solving the original problem. The SDP relaxation of the QAP subject to the decision variable being a permutation matrix given in [124] is:

$$\begin{aligned}\mu^* = \underset{X}{\arg\min} \quad & \mathrm{Tr}(AXBX^T - 2CX^T) \\ \text{s.t.} \quad & XX^T = X^TX = I && \text{(6.55a)}\\ & \|X\mathbf{1} - \mathbf{1}\|_2^2 + \|X^T\mathbf{1} - \mathbf{1}\|_2^2 = 0 && \text{(6.55b)}\\ & X_{ij}^2 - X_{ij} = 0, \quad \forall i,j && \text{(6.55c)}\end{aligned}$$

In this optimization problem (6.55a) is the now familiar orthogonal constraint, constraint (6.55b) captures the zero row and column sum constraint. Finally

constraint (6.55c) captures the $(0, 1)$ binary entry constraint. The optimization problem above is of the same form as (6.48) presented at the beginning of this section. In order to obtain a feasible solution the dual of (6.55) must be solved. The exact details of how this can be done are described in [123, 124].

6.7 Conclusion

In this chapter we have derived stability criteria for systems connected in feedback using notions from dissipative systems theory. The stability measures derived are of particular use when the subsystems have been produced from a decomposition algorithm. It was then shown how a decomposition based approach can, in conjunction with dissipation inequalities be used to estimate the $\mathcal{L}_2 \rightarrow \mathcal{L}_2$ gain of a large-scale nonlinear dynamical system.

The second half of the chapter was concerned with deriving bounds on the $\mathcal{L}_2$ norm of the output signal of LTI systems when extra states are appended to the system. It was shown that when a state (or set of states) is added down-stream of the existing system then a bound can be computed in a distributed manner as there is no coupling in the variables. Methods were also presented for cases corresponding to states being appended to up-stream of, and fully coupled to an existing system. In each case we derived upper and lower bounds on the maximum achievable $\mathcal{L}_2$ norm of the output. Finally, based on these results an algorithm that decomposes a system via state-aggregation or clustering was outlined.

Future work in this direction will involve deriving theoretical properties of the clustering algorithm and also incorporating the stability measures derived in the first part of the chapter into the algorithm.

6.8 Appendix

A.1 Storage functions

The storage functions and multipliers (to 3 s.f.) corresponding to example (6.22) are given below:

$$\begin{aligned} V_1 \;=\; & 1.04x_3x_1x_2 + 1.66x_3^2 + 1.57x_1x_2 - 1.21x_3^2x_1 + 0.568x_2x_3 + 648x_3^2x_1^2 \\ & +1.09x_1^2x_3 + 0.543x_3^3x_1 - 0.247x_2^2x_1 + 1.26x_3x_1^3 + 0.614x_3^3 - 0.672x_1^2x_2 \\ & -1.34x_1x_3 - 1.67x_3^2x_2 + 0.878x_3x_2^2 - 0.000574x_3^2x_2^2 + .255x_1^4 + 1.6x_2^2 \\ & +1.57x_1^2 - 0.343x_2^3 - 0.205x_1^3 + 1.05x_3^4 \end{aligned}$$

$$V_2 \;=\; 0.173x_4^2 - 0.0377x_4x_5 + 0.806x_5^2 + 0.0108x_4x_6 + 0.00232x_5x_6 + 0.204x_6^2.$$

The multipliers α_1, α_2 that make $V_c(x) = \alpha_1 V_1 + \alpha_2 V_2$ a composite storage function for system (6.22) are $\alpha_1 = 0.366$ and $\alpha_2 = 0.778$.

A.2 KYP Lemma

The *KYP lemma* is used to show the equivalence between the $\mathcal{H}_\infty$ norm of an LTI system of the form $G = C(sI - A)^{-1}B + D$ in the frequency domain and the existence of a feasible solution to an LMI involving the state space realization of G. There are many versions of this result, here we follow the presentation given in [17]:

Lemma 8 *Let $G = C(sI - A)^{-1}B + D$. Then the following are equivalent conditions:*

1. *The matrix A is Hurwitz and*

$$\|G\|_{\mathcal{H}_\infty} < 1;$$

2. *There exists a matrix* $P \succ 0$ *such that*

$$\begin{bmatrix} C^T \\ D^T \end{bmatrix} \begin{bmatrix} C & D \end{bmatrix} + \begin{bmatrix} A^T P + PA & PB \\ B^T P & -I \end{bmatrix} \prec 0.$$

A.3 Matrix Congruence, products and simultaneous diagonalization

Unlike the positive (real) scalar case, matrix multiplication of two positive definite matrices does not preserve positive definiteness. The following theorem is required in order to prove Lemma 5. Recall that a matrix is *Hermitian* if it is square and equal to its own conjugate transpose. Given two square matrices $A, B \in \mathcal{F}^n$ where $\mathcal{F}^n$ denotes a field, A and B are said to be congruent to each other if there exists a matrix $C \in \mathcal{F}^n$ such that $B = C^* AC$ where $*$ denotes the conjugate transpose. We denote the set of complex valued $n \times n$ matrices by $\mathcal{C}^{n \times n}$.

Theorem 15 ([30]) *The product of a positive definite matrix* $A \in \mathcal{S}^n_{++}$ *and a Hermitian matrix* $B \in \mathcal{C}^{n \times n}$ *is a diagonalizable matrix, all of whose eigenvalues are real. The matrix* AB *has the same number of positive, negative and zero eigenvalues as* B*. Furthermore, any diagonalizable matrix with real eigenvalues is the product of a positive definite matrix and a Hermitian matrix.*

The proof relies on the fact that A and B are congruent matrices. In the case of Lemma 5, B is negative definite and so it must be true that all eigenvalues of AB are real and negative.

Chapter 7

Conclusions

In this thesis we have introduced a *decomposition* framework for the scalable analysis of dynamical systems described by ordinary differential equations. The methods presented are formulated as convex optimization algorithms and use Lyapunov and Storage function arguments. In the first instance Linear Time Invariant (LTI) dynamical systems were analyzed using semidefinte programming before moving onto nonlinear analysis via Sum of Squares programming. The decomposition problem was considered from both a top-down and bottom up perspective.

To conclude, we briefly summarize the main contributions of each chapter and outline some interesting possible future research directions. In particular, some technical details regarding optimization over of the space of permutation matrices that may prove useful are given.

7.1 Summary

The following results have been presented in this thesis:

- In Chapter 3, a method for parameter estimation for nonlinear models using sum of squares programming was presented. We then investigated the problem of model invalidation in the presence of observed data. In the

continuous time case the main result was a re-parameterization of the time variable in the barrier certificate theorem. The new formulation was shown to improve the numerical conditioning of the resulting sum of squares optimization problem. We then turned our attention to the discrete-time invalidation problem to which a novel solution based on the positivstellensatz was formulated.

- In Chapter 4, the system decomposition problem was formulated and an energy-based algorithm was described that sought to facilitate the construction of a composite Lyapunov function. We first investigated decomposition for the LTI system class before proceeding to examine the nonlinear dynamical system case using sum of squares methods. One of the biproducts of this work is a numerical method for constructing sparse Lyapunov functions. The methods were presented on nonlinear models of an ecological network and a primal Internet congestion control scheme.

- Chapter 5 presented a detailed application of the nonlinear decomposition algorithm on a 35-state model of the EGF-MAPK signaling pathway. It was shown that the automated decomposition algorithm was able to uncover biologically significant subsystems in a complex network thus validating the energy-flow metric derived in the previous chapter. The concept of subnetwork discovery was then expanded upon to take into account nonlinear models containing uncertainty that could be represented by a Linear Parameter Varying (LPV) system model.

- In Chapter 6, based on the observation that the most computationally intensive stage of the decomposition framework is the verification step at the end (verifying the negativity of the derivative of the Lyapunov function) we investigated stability methods using the theory of dissipative dynamical systems. We investigated stability certificates based on the small gain theorem and various measures of passivity. Following on from this we then derived

bounds on the magnitude of the output of a system when states are appended up and down-stream of a system. Using these bounds we proposed that a bottom-up clustering based algorithm may be a viable alternative to the top-down approach presented previously.

7.2 Future research directions

As for future research directions, it would be interesting to extend the decomposition algorithm to take into account systems with inputs and outputs. A first attempt in this direction was presented in Section 6.3 in the context of estimating the $\mathcal{L}_2 \to \mathcal{L}_2$ gain of a system. What would be desirable however, is to formulate a decomposition algorithm that actually takes into account the input and output maps when computing the state decomposition. Such an approach could be thought of as being analogous to the model reduction technique of *balanced truncation*.

Throughout this thesis it has been emphasized that designing scalable algorithms for stability and performance analysis of nonlinear dynamical systems is a non-trivial task. The task of controller synthesis is even more challenging, and as Parrilo noted [8], "*extending this procedure to the nonlinear case does not seem feasible, at least in a reasonably straightforward way*". There is a clear link to controller synthesis related to the work in this thesis, specifically decentralized and/or distributed control. Designing a controller for even a large-scale linear system can be problematic. Decomposition methods may partially alleviate the curse of dimensionality but introduce the problem of non-convexity. For the case of nonlinear systems this non-convexity problem is already present. Recently for linear decentralized control synthesis a technique based on partially ordered set or *posets* has been introduced that uses directed graph theoretical notions [125]. We believe that there is a strong link between decomposition and poset causal systems that is worth investigating.

Another potentially interesting analysis question involves singularly perturbed systems [126]. Frequently, nonlinear dynamical systems contain states that evolve over multiple time-scales. Such systems are often referred to as *stiff* systems. To the best of our knowledge there does not exist a sum of squares (or alternative) method that takes this time scale into consideration when constructing Lyapunov functions algorithmically. It may prove useful to investigate techniques for automatically decomposing a system based on temporal state evolution in order to facilitate stability analysis.

Finally what remains to be done is to establish the technicalities of the proposed clustering algorithm in Chapter 6 and derive some theoretical results for such an approach. While the algorithm in its current state is presented in the context of the small gain theorem, the stability metrics described in Section 6.4 could potentially also be included. Note however that this is still likely to be a combinatorial problem and some form of convex relaxation will necessarily be required to make it tractable for large systems.

Appendix A

Notation

$\mathbb{Z}$	The field of integers
$\mathcal{R}$	The field of reals
$\mathcal{R}_{++}$ ($\mathcal{R}_{+}$)	The set of positive (non-negative) reals
$\mathcal{R}^n$	The set of real n-vectors
$\mathcal{R}^{n\times m}$	The set of real $n \times m$ matrices
$\mathcal{S}^m$	The set of $m \times m$ symmetric matrices
$\mathcal{C}^{n\times m}$	The set of complex $n \times m$ matrices
$\mathcal{S}^m_{++}$ ($\mathcal{S}^m_{+}$)	The set of $m \times m$ positive (semi-) definite matrices
$\mathcal{R}[x]$	The ring of polynomials in $x \in \mathcal{R}^n$ with real coefficients
$\Sigma[x]$	The set of SOS polynomials in $x \in \mathcal{R}^n$ of fixed degree
$\partial(p)$	Degree of $p(x) \in \mathcal{R}[x]$
Π_n	Group of $n \times n$ permutation matrices
$X \succ Y$ ($X \succeq Y$)	The matrix $X - Y$ is positive (semi-) definite
$\mathrm{Tr}(X)$	Trace of the matrix X
$\overline{\lambda}(Y)$, ($\underline{\lambda}(Y)$)	Maximum (minimum) eigenvalue value of the square matrix Y
$\overline{\sigma}(X)$, ($\underline{\sigma}(X)$)	Maximum (minimum) singular value of X
$[c]$	Concentration of species c
$\mathrm{card}(\mathcal{Z})$	Cardinality of the set $\mathcal{Z}$

References

[1] J. D. Murray, *Mathematical Biology*, 3rd ed., ser. Interdisciplinary Mathematics (17). Springer, 2002. 1, 35, 41, 42, 55

[2] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002. 1, 33

[3] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001. 2

[4] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, Inc., 2001. 3, 10, 11, 67, 114

[5] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994. 3, 12, 23, 29, 126, 129, 147

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. 3, 12, 13, 14, 71, 154

[7] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*. Berlin: Springer-Verlag, 1998. 3, 16, 21, 22

[8] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, Caltech, Pasadena, CA, 2000. 3, 8, 19, 163

[9] L. Vandenberghe and S. Boyd, "Semi-definite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996. 3, 14

[10] J. Anderson and A. Papachristodoulou, "On validation and invalidation of biological models," *BMC Bioinformatics*, vol. 10, no. 1, p. 132, 2009. 5

[11] ——, "A network decomposition approach for efficient sum of squares programming based analysis," in *Proc. of the American Control Conference*, 2010, pp. 4492–4497. 6

[12] ——, "Dynamical system decomposition for efficient, sparse analysis," in *Proc. of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6565–6570. 6

[13] ——, "A decomposition technique for nonlinear dynamical system analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1516–1521, June 2012. 6

[14] J. Anderson, Y.-C. Chang, and A. Papachristodoulou, "Model decomposition and reduction tools for large-scale networks in systems biology," *Automatica*, vol. 47, no. 6, pp. 1165 – 1174, 2011. 6

[15] J. Anderson, A. Teixeira, H. Sandberg, and A. Papachristodoulou, "Dynamical system decomposition using dissipation inequalities," in *Proc. of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2012, pp. 6516–6521. 6, 132

[16] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "SOSTOOLS – Sum of Squares Optimization Toolbox, User's Guide," http://sysos.eng.ox.ac.uk/sostools, 2002. 7, 20

[17] G. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*, ser. Texts in Applied Mathematics (36). Springer, 2000. 12, 81, 114, 147, 159

[18] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. Optim*, vol. 5, pp. 13–51, 1995. 14, 66

[19] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1994. 14

[20] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999. [Online]. Available: http://fewcal.kub.nl/sturm/software/sedumi.html 15

[21] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, 2011. 15

[22] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 – a matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999. 15

[23] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip 15

[24] K. G. Murty and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical Programming*, vol. 39, no. 2, pp. 11–129, 1987. 16

[25] B. Reznick, "Some concrete aspects of Hilbert's 17th problem," in *In Contemporary Mathematics*. American Mathematical Society, 1996, pp. 251–272. 18

[26] G. Blekherman, "There are significantly more nonegative polynomials than sums of squares," *Israel Journal of Mathematics*, vol. 153, pp. 355–380, 2006. 18, 32

[27] B. Reznick, "Uniform denominators in Hilbert's seventeenth problem," *Mathematische Zeitschrift*, vol. 220, pp. 75–97, 1995. 18

[28] V. Powers and T. Wörmann, "An algorithm for sums of squares of real polynomials," *Journal of Pure and Applied Algebra*, vol. 127, pp. 99–104, 1998. 19

[29] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, pp. 293–320, 2003. 19, 39

[30] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985. 20, 139, 160

[31] D. Cox, J. Little, and D. O'Shea, *Ideals, Varietis, and Algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer, 1997. 21

[32] A. L. Fradkov and V. A. Yakubovich, "The s-procedure and the duality relation in convex quadratic programming problems," *Vestnik Leningrad Univ. Math*, vol. 5, pp. 101 – 109, 1973. 23

[33] P. A. Parrilo and B. Sturmfels, "Minimizing polynomials functions," *arXiv:math/0103170v1*, 2001. 23, 39

[34] M. Fiedler, "Algebraic connectivity of graphs (english)," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, 1973. 23

[35] F. R. K. Chung, "Spectral graph theory," *CBMS 92, Amer. Math. Soc., Providence, 1997*, 1997. 23, 65, 105

[36] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, 2002, pp. 3482 – 3487. 26

[37] ——, "Analysis of non-polynomial systems using the sum of squares decomposition," in *Positive Polynomials in Control*, ser. Lecture Notes in Control and Information Sciences, D. Henrion and A. Garulli, Eds. Springer Berlin, 2005, vol. 312, pp. 580–580. 26, 35

[38] E. J. Hancock and A. Papachristodoulou, "Generalised absolute stability and sum of squares," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 2302–2307. 26

[39] G. Chesi, "Estimating the domain of attraction for uncertain polynomial systems," *Automatica*, vol. 40, no. 11, pp. 1981 – 1986, 2004. 26

[40] A. Papachristodoulou, M. M. Peet, and S. Lall, "Analysis of polynomial systems with time delays via the sum of squares decomposition," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1058–1064, 2009. 26

[41] A. Papachristodoulou, A. Jadbabaie, and U. Münz, "Effects of delay in multi-agent consensus and oscillator synchronization," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1471–1477, 2010. 26

[42] S. Prajna, A. Papachristodoulou, and F. Wu, "Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach," in *Proceedings of the 5th Asian Control Conference*, vol. 1, July 2004, pp. 157–165. 26

[43] S. Prajna, P. A. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 310–314, 2004. 26

[44] G. Chesi and D. Henrion, "Guest Editorial: Special Issue on Positive Polynomials in Control," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 935–936, 2009. 27

[45] M. M. Peet, "Exponentially stable nonlinear systems have polynomial lyapunov functions on bounded regions," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 979–987, May 2009. 27

[46] M. M. Peet and A. Papachristodoulou, "A converse sum-of-squares lyapunov result with a degree bound," *To Appear in IEEE Transactions on Control*, 2011. 27

[47] A. A. Ahmadi, M. Kristic, and P. A. Parrilo, "A globally asymptotically stable polynomial vector field with no polynomial lyapunov function," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 7579–7580. 27

[48] G. Chesi, "Estimating the domain of attraction for uncertain polynomial systems," *Automatica*, vol. 40, no. 11, pp. 1981 – 1986, 2004. 29

[49] U. Topcu, A. Packard, P. Seiler, and G. J. Balas, "Robust region-of-attraction estimation," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 137 –142, 2010. 29

[50] B. Tibken and Y. Fan, "Computing the domain of attraction for polynomial systems via BMI optimization method," in *Proceedings of the American Control Conference*, 2006, pp. 117–122. 29

[51] B. Reznick, "Extremal PSD forms with few terms," *Duke Mathematical Journal*, vol. 45, no. 2, pp. 363–374, 1978. 32, 70

[52] K. Gatermann and P. A. Parrilo, "Symmetry groups, semidefinite programs, and sums of squares," *Journal of Pure and Applied Algebra*, vol. 192, no. 1–3, pp. 95 – 128, 2004. 32, 70

[53] M. M. Peet and Y. Peet, "Solving large-scale robust control problems by exploiting the parallel structure of Polya's theorem," *Submitted to IEEE Transactions on Automatic Control*, pp. 1–33, 2011. 32

[54] U. Topcu, A. K. Packard, and R. M. Murray, "Compositional stability analysis based on dual decomposition," in *Proceedings of the 48th IEEE Conference Decision and Control*, 2009, pp. 1175 –1180. 32

[55] A. A. Ahmadi and P. A. Parrilo, "A complete characterization of the gap between convexity and sos-convexity," *Submitted to SIAM Journal on Optimization*, pp. 1–24, 2011. 32

[56] G. Dullerud and R. Smith, "A nonlinear functional approach to LFT model validation," *Systems & Control Letters*, vol. 47, no. 1, pp. 1 – 11, 2002. 34, 36

[57] D. G. Bates and C. Cosentino, "Validation and invalidation of systems biology models using robustness analysis," *Systems Biology, IET*, vol. 5, no. 4, pp. 229 –244, 2011. 34

[58] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall Inc, 1999. 36

[59] R. S. Smith and J. C. Doyle, "Model validation: A connection between robust control and identification," *IEEE Transactions on Automatic Control*, vol. 37, no. 7, pp. 942–952, 1992. 36

[60] M. P. Newlin and R. S. Smith, "A generalization of the structured singular value and its application to model validation," *IEEE Trans. Automat. Contr*, vol. 43, no. 7, pp. 901–907, 1998. 36

[61] R. Smith and G. E. Dullerud, "Continuous-time control model validation using finite experimental data," *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1094 –1105, 1996. 36

[62] L. Perko, *Differential equations and dynamical systems*, 3rd ed., ser. Texts in Applied Mathematics (7). Springer, 2000. 37

[63] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001. 39

[64] S. Prajna, "Optimization-based methods for nonlinear and hybrid systems verification," Ph.D. dissertation, Caltech, Pasadena, CA, 2005, available at http://etd.caltech.edu/etd/available/etd-05272005-144358/. 45

[65] ——, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 2, pp. 117–126, 2006. 45, 46, 48, 58

[66] S. Glavaski, A. Papachristodoulou, and K. Ariyur, "Safety verification of controlled advanced life support system using barrier certificates," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds. Springer Berlin / Heidelberg, 2005, vol. 3414, pp. 306–321. 47

[67] G. A. Shah, M. Völker, C. Sonntag, and S. Engell, "A barrier certificate approach to the verification of the safe operation of a chemical reactor," in *Proceedings of the 17th IFAC World Congress*, 2008, pp. 6932–6937. 47

[68] H. El-Samad, S. Prajna, A. Papachristodoulou, J. Doyle, and M. Khammash, "Advanced methods and algorithms for biological networks analysis," *Proceedings of the IEEE*, vol. 94, no. 4, pp. 832–853, 2006. 47

[69] P. Rumschinski, S. Borchers, S. Bosio, R. Weismantel, and R. Findeisen, "Set-base dynamical parameter estimation and model invalidation for biochemical reaction networks," *BMC Systems Biology*, vol. 4, no. 1, p. 69, 2010. 53

[70] M. Cannon, J. Buerger, B. Kouvaritakis, and S. Rakovic, "Robust tubes in nonlinear model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1942 –1947, 2011. 57

[71] F. N. Bailey, "The application of Lyapunov's second method to interconnected systems," *J. SIAM Control*, vol. 3, no. 3, pp. 443–462, 1966. 63, 64

[72] M. Araki, "Stability of large-scale nonlinear systems – quadratic-order theory of composite-system method using M-matrices," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 129–142, 1978. 63

[73] E. Kaszkurewicz and A. Bhaya, "Robust stability and diagonal Lyapunov functions," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 2, pp. 508–520, 1993. 63

[74] P. J. Moylan and D. J. Hill, "Stability criteria of large-scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 143–149, 1978. 63, 110, 111, 113

[75] J. C. Willems, "Dissipative dynamical systems. II. Linear systems with quadratic supply rates," *Arch. Rational Mech. Anal.*, vol. 45, pp. 352–393, 1972. 63, 110, 111

[76] J. Xiang and G. Chen, "On the V-stability of complex dynamical networks," *Automatica*, vol. 43, pp. 1049–1057, 2007. 63

[77] D. D. Šiljak, *Large-scale dynamic systems: Stability and Structure*. North-Holland, New York, 1978. 63

[78] M. Ikeda and D. D. Šiljak, "Generalized decompositions of dynamic systems and vector Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1118 – 1125, 1981. 63

[79] F. Callier, W. Chan, and C. Desoer, "Input-output stability of interconnected systems using decompositions: An improved formulation," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 150 – 163, 1978. 64

[80] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer-Verlag, Inc., 2000. 64

[81] R. Diestel, *Graph Theory*, 4th ed., ser. Graduate Texts in Mathematics. Heidelberg: Springer Verlag, 2010, vol. 173. 64

[82] H. Wolkowicz and Q. Zhao, "Semidefinite programming relaxations for the graph partitioning problem," *Discrete Applied Mathematics*, vol. 96-97, pp. 461 – 479, 1999. 66

[83] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998. 66

[84] S. E. Karisch, F. Rendl, and J. Clausen, "Solving graph bisection problems with semidefinite programming," *INFORMS J. on Computing*, vol. 12, pp. 177–191, July 2000. 66

[85] G. P. Barker, A. Berman, and R. J. Plemmons, "Positive diagonal solutions to the lyapunov equations," *Linear and Multilinear Algebra*, vol. 5, no. 4, pp. 249–256, 1978. 69

[86] R. Shorten, O. Mason, and C. King, "An alternative proof of the Barker, Berman, Plemmons (BBP) result on diagonal stability and extensions," *Linear Algebra and its Applications*, vol. 430, no. 1, pp. 34 – 40, 2009. 69

[87] T. Tanaka and C. Langbort, "The bounded real lemma for internally positive systems and H-infinity structured static state feedback," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2218 –2223, 2011. 69

[88] C. Langbort and V. Ugrinovskii, "Diagonal stability of stochastic systems subject to nonlinear disturbances and diagonal H_2 norms," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 3188–3193. 69

[89] R. Cogill, S. Lall, and P. A. Parrilo, "Structured semidefinite programs for the control of symmetric systems," *Automatica*, vol. 44, no. 5, pp. 1411 – 1417, 2008. 70

[90] J. Lofberg, "Pre- and post-processing sum-of-squares programs in practice," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1007–1011, 2009. 70

[91] S. Boyd and Q. Yang, "Structured and simultaneous Lyapunov functions for system stability problems," *International Journal of Control*, vol. 49, no. 6, pp. 2215–2240, 1989. 74

[92] H. K. Khalil and P. V. Kokotovic, "D-stability and multi-parameter singular perturbation," *SIAM Journal on Control and Optimization*, vol. 17, no. 1, pp. 56–65, 1979. 75

[93] D. G. Feingold and R. S. Varga, "Block diagonally dominant matrices and generalizations of the Gerschgorin Circle Theorem," *Pacific J. Math*, vol. 12, pp. 1241–1250, 1962. 76, 142

[94] F. Kittaneh, "Spectral radius inequalities for Hilbert space operators," *Proceedings of the American Mathematical Society*, vol. 134, no. 2, pp. 385–390, 2005. 78

[95] M. E. J. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, pp. 8577–8582, 2006. 80

[96] ——, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, 2006. 80

[97] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, 2004. 80

[98] J. Scherpen, "Balancing for nonlinear systems," *Systems & Control Letters*, vol. 21, no. 2, pp. 143 – 153, 1993. 81

[99] S. Prajna and H. Sandberg, "On model reduction of polynomial dynamical systems," in *Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference.*, 2005, pp. 1666 – 1671. 81

[100] S. Lall, J. E. Marsden, and S. Glavaški, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 6, pp. 519–535, 2002. 84

[101] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998. 84

[102] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhäuser, 2003. 84, 85

[103] M. Ikeda and D. D. Šiljak, "Lotka-Volterra equations: Decomposition, stability, and structure," *Journal of Mathematical Biology*, vol. 9, pp. 65–83, 1980. 87

[104] A. Halanay and V. Răsvan, *Applications of Lyapunov Methods in Stability*. Kluwer Academic Press, 1993. 87

[105] M. Ikeda and D. D. Šiljak, "Generalized decompositions of dynamic systems and vector Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1118 – 1125, 1981. 90

[106] M. Ikeda, D. D. Šiljak, and D. E. White, "Decentralized control with overlapping information sets," *Journal of Optimization Theory and Applications*, vol. 34, pp. 279–310, 1981. 90

[107] S. Dutt, "New faster Kernighan-Lin-type graph-partitioning algorithms," in *Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD '93, 1993, pp. 370–377. 90

[108] G. Ziegler, *Lectures on Polytopes*. Springer, 1995. 93

[109] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. Willsky, "The convex geometry of linear inverse problems," December 2010, under review, available online at http://ssg.mit.edu/~venkatc/crpw_lip_preprint10.pdf. 93

[110] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of Process Control*, vol. 10, no. 4, pp. 363 – 385, 2000. 93

[111] B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Muller, "Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors," *Nat Biotech*, pp. 370–375, 2002. 98, 99

[112] J. Saez-Rodriguez, A. Kremling, H. Conzelmann, K. Bettenbrock, and E. Gilles, "Modular analysis of signal transduction networks," *Control Systems Magazine, IEEE*, vol. 24, no. 4, pp. 35–52, 2004. 99

[113] H. Conzelmann, J. Saez-Rodriguez, T. Sauter, E. Bullinger, F. Allgower, and E. Gilles, "Reduction of mathematical models of signal transduction networks: simulation-based approach applied to EGF receptor signalling," *IEE Systems Biology*, vol. 1, no. 1, pp. 159 –169, June 2004. 99, 101

[114] J. Saez-Rodriguez, S. Gayer, M. Ginkel, and E. D. Gilles, "Automatic decomposition of kinetic models of signalling networks minimizing the retroactivity among modules," *Bioinformatics*, vol. 24, pp. i213–i219, 2008. 101

[115] A. Varga, G. Moormannand, and G. Grubel, "Automated generation of LFT -based parametric uncertainty descriptions from generic aircraft models," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 4, no. 4, pp. 249–274, 1998. 103

[116] P. Gahinet, P. Apkarian, and M. Chilali, "Affine parameter-dependent Lyapunov functions and real parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 436–442, 1996. 104, 106

[117] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1995. 114

[118] A. J. van der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770 –784, 1992. 115

[119] S. Dashkovskiy, B. Rüffer, and F. R. Wirth, "An ISS small gain theorem for general networks," *Mathematics of Control, Signals, and Systems*, vol. 19, pp. 93–122, 2007. 126

[120] H. Sandberg, "Hankel-norm-based lumping of interconnected linear systems," in *Proc. of MATHMOD*, 2009, pp. 1278–1286. 133

[121] K. Yasuda and K. Hirai, "Upper and lower bounds on the solution of the algebraic Riccati equation," *IEEE Transactions on Automatic Control*, vol. 24, no. 3, pp. 483–487, June 1972. 138

[122] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991. 139

[123] K. Anstreicher and H. Wolkowicz, "On lagrangian relaxation of quadratic matrix constraints," *SIAM J. Matrix Anal. Appl*, vol. 22, pp. 41–55, 1998. 155, 157, 158

[124] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz, "Semidefinite programming relaxations for the quadratic assignment problem," *Journal of Combinatorial Optimization*, vol. 2, pp. 71–109, 1998. 155, 157, 158

[125] P. Shah and P. A. Parrilo, "H2-optimal decentralized control over posets: A state space solution for state-feedback," in *In proc. of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6722–6727. 163

[126] P. Kokotović, H. K. Khalil, and J. O'Reilly, *Singular perturbation methods in control: Analysis and design*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999. 164