Large-Scale Layered Systems and Synthetic Biology: Model Reduction and Decomposition

Thomas P. Prescott

Department of Engineering Science

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas Term 2014

Acknowledgements

This thesis would have been impossible without the support of my supervisor, Professor Antonis Papachristodoulou, whose advice, enthusiasm, encouragement, and relentless idea-generation have kept me motivated and inspired throughout this project. He has built a group of intimidatingly clever, successful students and postdocs, whose diverse interests and creativity have ensured that my DPhil experience has taught me far more than any number of solitary reading sessions ever could. Special mentions go to Richard Mason, Xuan Zhang, Ed Hancock, and James Anderson for their support and advice, right from the start.

Outside of Antonis' group, my office-mates Matt, Giacomo, Ingrid and Akis have provided invaluable help and tea-time distraction. My collaborators, Moritz Lang at ETH and Andras Gyorgy at MIT, have also both been great fun to work with. Many thanks to their supervisors, Jörg Stelling and Domitilla Del Vecchio, for their hospitality during my visits. I am especially grateful to Jörg and to Philip Maini for agreeing to be my examiners, and to Basil Kouvritakis and David Limebeer for being on my Transfer committee.

I gratefully acknowledge the support given by EPSRC through the Life Sciences Interface DTC (and a number of Antonis' grants), and by Trinity College. For providing opportunities to teach and interview undergraduates, thanks are due to: Elspeth Garman from the DTC; Mason Porter from the Maths Institute; and the Maths tutors of Trinity.

An important reason I've not been beaten by this thesis is due the support, wisdom, and perspective of my family and friends. I look forward year-round to Liverp-yule and seeing Katie and Alex, the Paloes, and the Traffs. Visits to the Walters–Constable family have also been a welcome break from Oxford, and I'm deeply indebted to their hospitality. From the Golden Boys and Old Boys, and from JO (and WO) and Phil, to the FloMa Alum. Assoc., *les Fettlers*, and countless Common Room iterations, I've been extremely fortunate to have such fantastic friends. Very special mentions must also go to the ex-residents of Gillman House (and associated hangers-on), and to all parties to the Sherry Trifle Dispute. I should also mention James Rawson (although/because he has been indirectly mentioned three times already), for whose friendship I'm eternally grateful. Finally I should thank Tom Mayo and Matt Lawes, for having been magnificent friends and stalwart supports for the past decade.

Of course, as is conventional, these acknowledgements are for those who helped me to write this thesis. However, I have been learning for a much longer time than that taken to complete my DPhil, and there are many more people, not mentioned above, to whom I am gratefully indebted.

Most importantly, my thanks go to my parents, Ann and Paul, and to my sister Katie. This DPhil directly results from the value they place on hard work, loyalty, dedication, and education. And my final thanks are reserved for the Lovely Kate Walters, whose patience, humour, support, and love cannot possibly be overstated.

Thomas P. Prescott
Oxford, January 2015
A. M. D. G.

Abstract

Large-Scale Layered Systems and Synthetic Biology: Model Reduction and Decomposition

Thomas P. Prescott
Trinity College
Submitted for the degree of *Doctor of Philosophy* in Michaelmas Term 2014

This thesis is concerned with large-scale systems of Ordinary Differential Equations that model Biomolecular Reaction Networks (BRNs) in Systems and Synthetic Biology. It addresses the strategies of model reduction and decomposition used to overcome the challenges posed by the high dimension and stiffness typical of these models. A number of developments of these strategies are identified, and their implementation on various BRN models is demonstrated.

The goal of model reduction is to construct a simplified ODE system to closely approximate a large-scale system. The *error estimation problem* seeks to quantify the approximation error; this is an example of the *trajectory comparison problem*. The first part of this thesis applies semi-definite programming (SDP) and dissipativity theory to this problem, producing a single *a priori* upper bound on the difference between two models in the presence of parameter uncertainty and for a range of initial conditions, for which exhaustive simulation is impractical.

The second part of this thesis is concerned with the *BRN decomposition problem* of expressing a network as an interconnection of subnetworks. A novel framework, called *layered* decomposition, is introduced and compared with established modular techniques. Fundamental properties of layered decompositions are investigated, providing basic criteria for choosing an appropriate layered decomposition. Further aspects of the layering framework are considered: we illustrate the relationship between decomposition and scale separation by constructing singularly perturbed BRN models using layered decomposition; and we reveal the inter-layer signal propagation structure by decomposing the steady state response to parametric perturbations.

Finally, we consider the *large-scale SDP problem*, where large scale SDP techniques fail to certify a system's dissipativity. We describe the framework of *Structured Storage Functions (SSF)*, defined where systems admit a cascaded decomposition, and demonstrate a significant resulting speed-up of large-scale dissipativity problems, with applications to the trajectory comparison technique discussed above.

Contents

Chapter 1

Introduction

This chapter outlines the scope of this thesis, which is to investigate two related techniques used in Control and Systems Engineering to analyse large, complicated models of dynamical systems. The thesis will focus in particular on models that arise in Systems and Synthetic Biology; Section 1.1 introduces and derives the Ordinary Differential Equation (ODE) models describing the dynamics of large-scale biomolecular reaction networks. The first of the two techniques is *model reduction*. Section 1.2 reviews the methods by which the behaviour of a detailed model can be approximated by a simpler model. The second technique, which is the topic of Section 1.3, is *model decomposition*, whereby the analysis of a detailed ODE model is split into the hierarchical analysis of multiple simpler models and their interconnection. The chapter concludes in Section 1.4 with an outline of the contribution of this thesis, posing a number of questions relating to the reduction and decomposition of biomolecular reaction network models.

1.1 Modelling Biomolecular Reaction Networks

The biological processes that take place in evolved organisms occur at many different spatial and temporal scales, ranging from the ecological dynamics of many interacting species of multicellular organisms to the flow of metabolites and signalling molecules within a cell [134, 219]. This thesis is concerned with the processes at the smaller end of this spectrum, focusing on the dynamics of how DNA, RNA and protein molecules interact with one another to result in the characteristic properties of a living cell. The quantitative understanding of these *biomolecular reaction networks* (BRNs) is aided by the construction of mathematical models, which enable the simulation, analysis, and control of biological systems [43, 51, 133].

There has been a great deal of research that exploits large-scale experimental measurement techniques to elucidate the various biomolecular species and their interactions within biological cells [52, 137, 216]. These techniques can be used to construct models that reflect the BRN's structure [11, 104, 190] and also to quantify particular parameter values [157]. The experiments must be designed to ensure that the parameters being estimated are in fact identifiable [66, 160], and also so that they can appropriately invalidate candidate models [9, 115]. The species whose dynamics are being modelled can be proteins, metabolites, various types of RNA, genetic promoters or inhibitors, and so on. Web-based '-omics' databases catalogue, for different model organisms, many types of cellular networks (genetic regulation, signalling, metabolic, protein interaction, etc). The number of these databases has been growing since the early years of this century [193]. Novel techniques are required to integrate these network layers together in order to generate large-scale models that fully characterise a cell [77, 86, 192, 194, 209] . However, in contrast to these large-scale efforts, many researchers tend to limit the scope of their network reconstruction to smaller functions within the cell.

In this thesis we will consider questions relating to the reduction and decomposition of Ordinary Differential Equation (ODE) models of BRNs. The model whose derivation we sketch below is a widely-used framework for the mathematical treatment of BRNs, and can be found in a number of textbooks [97, 137, 138].

1.1.1 Notation and derivation

Consider a BRN consisting of N chemical species X_i for $i = 1,\dots,N$, and denote the concentration of X_i at any time-point $t \in \mathbb{R}_{\geq 0}$ by the scalar value $x_i(t) \in \mathbb{R}_{\geq 0}$. Note that the units of concentration and time will be assumed, although not explicitly given, as a property of the model. The goal of the model derivation is to construct a system of N ODEs to track the trajectory of the *state vector* $\boldsymbol{x}(t) = \left[x_1(t),\dots,x_N(t)\right]^T \in \mathbb{R}^N_{\geq 0}$ from the initial concentrations $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

The BRN has M reactions, each of the form

$$\alpha_{1j}X_1 + \cdots + \alpha_{Nj}X_N \xrightarrow{v_j(\boldsymbol{x}(t))} \beta_{1j}X_1 + \cdots + \beta_{Nj}X_N, \tag{1.1}$$

for $j = 1,\dots,M$. We assume that the coefficients $\alpha_{ij}, \beta_{ij} \in \mathbb{Z}_{\geq 0}$ are non-negative integers, corresponding to the number of molecules of species X_i consumed and produced, respectively, as reaction j proceeds[1]. The net effect of reaction j is therefore to increase the number of molecules of X_i by the *stoichiometric coefficient*, defined as $S_{ij} = \beta_{ij} - \alpha_{ij} \in \mathbb{Z}$.

Each $v_j\big(\boldsymbol{x}(t)\big) \in \mathbb{R}_{\geq 0}$ corresponds to the rate (in dimensions of concentration over time) at which reaction j takes place, known as the *flux*. Note that the flux takes the argument $\boldsymbol{x}(t)$ because, at each time point $t \geq 0$, the reaction rate v_j will, in general, depend on the concentrations $x_i(t)$ of any of the N species. Furthermore, we assume that $v_j(\boldsymbol{x})$ cannot become negative. However, in practice many BRNs contain reversible reactions where, for example, a bound pair of molecules can unbind. Reversible reactions can still be modelled in this framework by simply defining two reactions with non-negative reaction rates. Reaction j represents the forward reaction at rate $v_j(\boldsymbol{x})$, and reaction $k \neq j$ represents the backward reaction at rate $v_k(\boldsymbol{x})$. The associated stoichiometries of these reactions are such that $S_{ij} = -S_{ik}$, for all $i = 1,\dots,N$. This modelling convention ensures that the *flux vector* $\boldsymbol{v}\big(\boldsymbol{x}(t)\big) \in \mathbb{R}^M_{\geq 0}$ always takes values in the non-negative orthant of $\mathbb{R}^M$. This is useful for computational approaches to analysing the steady-state flux distribution of the BRN [137].

The effect of reaction j proceeding at rate v_j is to change each concentration x_i at a rate equal to the product $S_{ij}v_j$ of the reaction rate with the stoichiometric coefficient S_{ij}. Aggregating the contributions of the full set of reactions $j = 1,\dots,M$ on species i, the total rate of change

[1] The specific assumption of integer values here is not strictly necessary for any of the results in this thesis.

of x_i is therefore given by the ODE

$$\dot{x}_i(t) = \sum_{j=1}^{M} S_{ij} v_j\big(\boldsymbol{x}(t)\big),$$

for each $i = 1, \ldots, N$. This collection of coupled ODEs is usually written in the compact form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{S}\boldsymbol{v}\big(\boldsymbol{x}(t)\big), \tag{1.2}$$

where the state vector $\boldsymbol{x}(t) \in \mathbb{R}^N_{\geq 0}$ was defined above, and the flux vector is the vector $\boldsymbol{v}\big(\boldsymbol{x}(t)\big) = \big[v_1\big(\boldsymbol{x}(t)\big), \ldots, v_M\big(\boldsymbol{x}(t)\big)\big]^T \in \mathbb{R}^M_{\geq 0}$ of fluxes. The matrix $\boldsymbol{S} = [S_{ij}] \in \mathbb{Z}^{N\times M}$ of stoichiometric coefficients S_{ij} is termed the *stoichiometric matrix.*

The functional form of each reaction rate $v_j(\boldsymbol{x})$ also needs to be identified [52]. A typical choice is to use the Law of Mass Action, where the reaction rate v_j is modelled by

$$v_j(\boldsymbol{x}) = k_j \prod_{i=1}^{N} (x_i)^{\alpha_{ij}}, \tag{1.3a}$$

where α_{ij} is given by the stoichiometry of (1.1), and the parameters k_j are positive constants, corresponding to reaction rates whose values must usually be derived experimentally. Other possible choices for v_j include assuming Michaelis–Menten kinetics [133], whereby the rate of the reaction $X_1 \xrightarrow{v} X_2$ is modelled by

$$v(\boldsymbol{x}) = \frac{V_{max} x_1}{K_m + x_1}. \tag{1.3b}$$

Thus the reaction rate is linear in x_1, with gradient V_{max}/K_m, for small values of x_1, but saturates at V_{max} for larger values of x_1.

To construct the model (1.1), and hence the ODEs in (1.2), both the stoichiometric matrix $\boldsymbol{S}$ and the form and parameter values of the flux vector $\boldsymbol{v}(\boldsymbol{x})$ need to be determined. As described above, there is a vast literature on model construction [137]. However, in this thesis we will assume that the ODE model (1.2) of a BRN, defined by an experimentally-deduced $\boldsymbol{S}$ and $\boldsymbol{v}(\boldsymbol{x})$, is given and reliable.

1.1.2 ODE models in Systems Biology

The Systems Biology Markup Language (SBML) [93] is an XML-based file format that allows researchers to define their model in terms of the structures $\boldsymbol{x}$, $\boldsymbol{v}$, and $\boldsymbol{S}$, and therefore enables standardised databases where models can be curated and downloaded [117]. Associated software, such as the SBTOOLBOX for MATLAB [174], can be used for model simulation. In addition to simulation, the analysis of ODE models of the form (1.2) can give a great deal of insight into the underlying biological processes. For example, algebraic properties of the stoichiometric matrix $\boldsymbol{S}$ give information about the behaviour of the BRN [137], such as defining conservation equations [101, 206] and steady state flux distributions. Many Systems Biology approaches to analysing the model (1.2) are methods for investigating the steady state, such as Flux Balance Analysis [102], Elementary Flux Modes [176, 177, 195], and forms of parameter sensitivity analysis [218] such as Metabolic Control Analysis [65], and related 'control analysis' methods [32, 89, 178]. Other properties of BRN dynamics, such as multi-stability [136, 173] and monotonicity [13, 14], can be deduced from (1.2).

As stated above, the ODE model (1.2) will be the topic of investigation in the remainder of this thesis. However, it is important to note some limitations to this model, and also alternative modelling frameworks. The reactions (1.1) define the edges of a hypergraph [107] on N nodes, each of which corresponds to a species X_i. A slightly different formulation of (1.2) is given by the representation originally described in [91]. This instead considers each reaction (1.1) as an edge between the nodes $\sum \alpha_{ij} X_i$ and $\sum \beta_{ij} X_i$, thereby defining a weighted directed graph. This representation enables the analysis of the network using various techniques from graph theory and network science, which together forms the literature in Chemical Reaction Network Theory (CRNT) [79]. Other graph-theoretic and combinatorial approaches to modelling biomolecular systems are also widespread [3, 27, 120, 168, 179].

The ODE models for BRNs constructed above make a number of assumptions on the biomolecular system. Within the cell the numbers of various molecules taking part in some reactions may be very small, so that stochastic effects cause ODE models to deviate from experimentally-verified reality [144]. In this setting, noise becomes an important factor in the dynamics of BRNs, which many researchers advocate explicitly modelling [61, 78, 158, 210]. Numerical techniques such as the Gillespie algorithm [73] are widely-used for the simulation of noisy chemical processes [130], and there have also been recent developments of analytical techniques for quantifying the behaviour of stochastic BRNs [33, 80, 164].

1.2 Model Order Reduction and Error

The models described above can be constructed at the scale of an entire genome, which often results in large, unwieldy models. Even when researchers focus on modelling particular cellular functions in precise detail, the resulting ODE systems tend to be large, in terms of the numbers N and M of biomolecular species and reactions, respectively. However, the stoichiometric matrix $\boldsymbol{S} \in \mathbb{Z}^{N \times M}$ is usually sparse, since in typical BRNs each of the species takes part in a relatively small number of reactions [123]. Notwithstanding the sparsity of $\boldsymbol{S}$, large ODE systems are difficult to deal with for a number of reasons.

Recall the derivation of the functional form of each reaction rate $v_j(\boldsymbol{x})$; each reaction rate of the form (1.3a) has a reaction rate constant k_j, while the Michaelis–Menten form (1.3b) requires two parameters, V_{max} and K_m. In a BRN with M reactions, there are at least M parameters to be identified. If M is large, this may require an intensive effort in order to design and implement the experimental strategy necessary to identify the parameters [16]. Once the kinetic parameters in a large-scale model have been identified, the high dimension of the state space $\mathbb{R}^N$ introduces new difficulties. In particular, simulating the trajectory of all N species in a system can become difficult as N gets large. If only a subset of the species are of interest, then much of the computational cost of accurately simulating and storing the full-scale network is wasted. Large system models also tend to confuse intuition about systems-level interpretations, obscuring the insight that less precise models may give. For example, the analysis of glycolytic oscillations in a low-dimensional model [37, 41] is much simpler than would be possible in a more precise, large-scale model [94]. In common with more traditional applications of Control Theory, another difficulty associated with large-scale dynamical systems arises when designing controllers. Often, these have the same level of complexity as the system model. Therefore, reducing the model before the construction of a controller will aid the design process [15].

One important difficulty with the analysis of full-scale BRNs is the stiffness in many of the associated ODE models, which arises from the multiple timescales that are characteristic of biological systems [99, 134]. When only a subset of the species is of interest, a lot of the computational burden incurred by simulating the full-scale model can be avoided, as many of the processes that occur on fast or slow timescales may not need to be modelled in full mechanistic detail in order to capture the dynamics of the system at a given timescale. Furthermore, simulating systems at multiple timescales requires specialised numerical techniques to deal with the stiffness of the system, adding further complications to the large-dimensional problem described above [38]. Hence, reduction techniques that can overcome model stiffness are particularly applicable to BRN models.

1.2.1 Model reduction problem statement

The model reduction problem is concerned with the reduction in dimension of an ODE system [15]. Suppose that there is a dynamical system of interest, which we call the *full-scale model* Γ. This is a map from a trajectory $\boldsymbol{u}$ and state $\boldsymbol{x}_0$ to a trajectory $\boldsymbol{y}$ defined by

$$\text{Full model, }\Gamma \quad \begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big), \\ \boldsymbol{y}(t) = \boldsymbol{h}\big(\boldsymbol{x}(t)\big), \\ \boldsymbol{x}(0) = \boldsymbol{x}_0, \end{cases}$$

with the input signal $\boldsymbol{u}$ taking values $\boldsymbol{u}(t) \in \mathscr{U} \subseteq \mathbb{R}^p$. In Γ, the state $\boldsymbol{x}$ takes values $\boldsymbol{x}(t) \in \mathscr{D} \subseteq \mathbb{R}^N$, and is initialised from an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ taking values in an initial condition set $\mathscr{X}_0 \subseteq \mathscr{D}$. The system Γ maps the inputs $\boldsymbol{u}$ and $\boldsymbol{x}_0$ to the measured output $\boldsymbol{y} := \Gamma(\boldsymbol{u}, \boldsymbol{x}_0)$ with values $\boldsymbol{y}(t) \in \mathbb{R}^q$.

Now suppose that a reduced dimension $\tilde{N} < N$ is defined. The model reduction task is to construct an $\tilde{N}$-dimensional *reduced model* $\tilde{\Gamma}$ with reduced state $\tilde{\boldsymbol{x}}$ taking values $\tilde{\boldsymbol{x}}(t) \in \mathbb{R}^{\tilde{N}}$. Mathematically, the task is to construct functions $\tilde{\boldsymbol{f}}$, $\tilde{\boldsymbol{h}}$, and $\boldsymbol{\chi}$ for the dynamics of the *reduced model*:

$$\text{Reduced model, }\tilde{\Gamma} \quad \begin{cases} \dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\boldsymbol{f}}\big(\tilde{\boldsymbol{x}}(t), \boldsymbol{u}(t)\big), \\ \tilde{\boldsymbol{y}}(t) = \tilde{\boldsymbol{h}}\big(\tilde{\boldsymbol{x}}(t)\big), \\ \tilde{\boldsymbol{x}}(0) = \tilde{\boldsymbol{x}}_0 = \boldsymbol{\chi}(\boldsymbol{x}_0), \end{cases}$$

such that the output $\tilde{\boldsymbol{y}} := \tilde{\Gamma}(\boldsymbol{u}, \boldsymbol{\chi}(\boldsymbol{x}_0))$, taking values $\tilde{\boldsymbol{y}}(t) \in \mathbb{R}^q$, well-approximates $\boldsymbol{y}$ in some sense. Note that it is important that the output $\tilde{\boldsymbol{y}}$ of the reduced model is of the same dimension, q, as that of the full model. Note also that the reduction requires a map $\boldsymbol{\chi}$ from the initial condition $\boldsymbol{x}_0 \in \mathscr{X}_0$ of the original model to the initial condition $\boldsymbol{\chi}(\boldsymbol{x}_0) \in \tilde{\mathscr{X}}_0$ of the reduced model, where $\tilde{\mathscr{X}}_0$ is the image of $\mathscr{X}_0$ under $\boldsymbol{\chi}$.

The criterion of 'well-approximates' can be quantified as follows. Denote as $\boldsymbol{y} = \Gamma(\boldsymbol{u}, \boldsymbol{x}_0)$ and $\tilde{\boldsymbol{y}} = \tilde{\Gamma}(\boldsymbol{u}, \boldsymbol{\chi}(\boldsymbol{x}_0))$ respectively the output trajectories $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ resulting from the full and reduced models with the input trajectory $\boldsymbol{u}$ and initial conditions $\boldsymbol{x}_0$ or $\boldsymbol{\chi}(\boldsymbol{x}_0)$ respectively. Suppose that there is a functional Δ of these two trajectories that takes values $\Delta\big(\Gamma(\boldsymbol{u},\boldsymbol{x}_0),\tilde{\Gamma}(\boldsymbol{u},\boldsymbol{\chi}(\boldsymbol{x}_0))\big) \in \mathbb{R}_{\geq 0}$. This functional is intended to define the difference or (in the context of model reduction) the *error* between the output trajectories $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$. It can be defined in a number of ways, depending on the property of the output that is important to the modeller. Therefore the model reduction problem is the task of finding a reduced model of dimension $\tilde{N}$, the choice of which may be constrained by physical or computational considerations, that well-approximates the full-scale model by, for example, minimising the largest (i.e. worst-case) value of $\Delta\big(\Gamma(\boldsymbol{u},\boldsymbol{x}_0),\tilde{\Gamma}(\boldsymbol{u},\boldsymbol{\chi}(\boldsymbol{x}_0))\big)$ across a particular space of inputs, such as $\boldsymbol{u}(t) \in \mathscr{U}$, and space of possible initial conditions $\boldsymbol{x}_0 \in \mathscr{X}_0$.

Reduction problems such as this have been studied in the extensive, established model reduction literature [15]. We can identify two important aspects of any solution: first, the construction of candidate reductions $\tilde{\Gamma}$ from Γ and, second, being able to calculate the error $\Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$ used as the optimal reduction criterion. This thesis will consider the commonly-used $\mathscr{L}_2$ error measure, defined by

$$\Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) = \left\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\right\|_2 = \left(\int_0^\infty |\boldsymbol{y}(t)-\tilde{\boldsymbol{y}}(t)|^2 \, \mathrm{d}t\right)^{1/2}. \tag{1.4}$$

for $\boldsymbol{y} = \Gamma(\boldsymbol{u},\boldsymbol{x}_0)$ and $\tilde{\boldsymbol{y}} = \tilde{\Gamma}(\boldsymbol{u},\tilde{\boldsymbol{x}}_0)$. To ensure a finite measure, this error norm requires an assumption on the stability of Γ and $\tilde{\Gamma}$ such that both $\boldsymbol{y}(t)$ and $\tilde{\boldsymbol{y}}(t)$ approach the same value $\boldsymbol{y}^\star = \tilde{\boldsymbol{y}}^\star$ as $t \to \infty$ (although incremental stability [12] may allow a relaxation of this assumption). This norm fits with many experimental conventions for the use of mean-squared error, and is also a natural choice for analysing the reduction of linear models. However, other error measures may be used in other contexts, such as the difference $\left|\boldsymbol{y}^\star - \tilde{\boldsymbol{y}}^\star\right|$ between the steady states, or other norms on the trajectory difference $\boldsymbol{y}-\tilde{\boldsymbol{y}}$.

1.2.2 Model reduction methods

In the context of non-autonomous linear systems, the model reduction problem with Δ given by (1.4) is known as the optimal $\mathscr{H}_\infty$ reduction problem, since it can be formulated as choosing $\tilde{\Gamma}$ to minimise the induced $\mathscr{L}_2$ to $\mathscr{L}_2$ gain of the *error system* (see Chapter 3). While this problem remains open in general [59], there are a number of methods for model reduction of linear systems [15]. One important reduction technique is balanced truncation [59, 63, 74]. The key to this approach is that there exists a state-space realisation of the input–output system Γ such that each state x_i is associated with a Hankel singular value σ_i that quantifies both its observability and controllability. Balanced truncation constructs $\tilde{\Gamma}$ by discarding the $N-\tilde{N}$ states of Γ with the smallest Hankel singular values. Ordering states such that $\sigma_1 \geq \cdots \geq \sigma_N > 0$, the induced error can be estimated by the bounds

$$\sigma_{\tilde{N}+1} \leq \sup_{\|u\|=1} \|\Gamma(\boldsymbol{u},\boldsymbol{0}) - \tilde{\Gamma}(\boldsymbol{u},\boldsymbol{0})\| \leq 2(\sigma_{\tilde{N}+1} + \cdots + \sigma_N),$$

assuming that the strict inequalities $\sigma_N < \sigma_{N-1} < \cdots < \sigma_{\tilde{N}+1} < \sigma_{\tilde{N}}$ hold. An analogous concept to balanced truncation in nonlinear input–output systems has also been defined [146, 172].

Another well-established model reduction method in Systems Theory and Mathematical Biology is singular perturbation [110, 175, 180], based on the following theorem.

Theorem 1.1 (Tikhonov's Theorem [108]). *Consider the coupled ODE system*

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{1.5a}$$

$$\mu\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{1.5b}$$

for an arbitrary parameter $\mu > 0$*, where the vector fields* $\boldsymbol{f}_i$ *are continuous, and where it is assumed that for given initial conditions* $\boldsymbol{x}_i(0) = \boldsymbol{x}_{i,0}$ *the ODEs have unique solutions. Define the map* $\boldsymbol{\phi}(\boldsymbol{x})$ *by the algebraic relationship* $\boldsymbol{0} = \boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{\phi}(\boldsymbol{x}))$*. Suppose that the system* $\dot{\boldsymbol{z}} = \boldsymbol{f}_2(\boldsymbol{x}, \boldsymbol{z})$ *for arbitrary fixed* $\boldsymbol{x}$ *has a stable equilibrium at* $\boldsymbol{z} = \boldsymbol{\phi}(\boldsymbol{x})$*. Now consider the approximated system*

$$\dot{\tilde{\boldsymbol{x}}}_1 = \boldsymbol{f}_1(\tilde{\boldsymbol{x}}_1, \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_1)), \tag{1.6a}$$

$$\tilde{\boldsymbol{x}}_2 = \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_1) \tag{1.6b}$$

from initial conditions $\tilde{\boldsymbol{x}}_1(0) = \boldsymbol{x}_{1,0}$*. As* $\mu \to 0$*, the solution of the full ODE system* (1.5) *is arbitrarily closely approximated by the solution of the reduced ODE system* (1.6).

This theorem can be used to construct and justify the model reduction of a given autonomous ODE model by singular perturbation. Given the autonomous dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$, the aim of this reduction technique is to identify a parameter $\mu \ll 1$ such that these dynamics can be written in the form (1.5). Assuming all other parameter values in the model are on the same scale, then small values of μ imply that the dynamics (1.6) well-approximate (1.5). However, unlike balanced truncation, there is no *a priori* $\mathscr{L}_2$ estimate of the error incurred by singular perturbation; this means that there is no criterion of exactly how small μ must be, given the rest of the model's parameters, to achieve a particular pre-defined error threshold. Nevertheless, there are many different approaches that have been taken in order to identify the most valid approximation [28, 48, 98, 113, 116, 133, 205].

Many model reduction techniques that are specific to biomolecular models (1.2) still tend to be based on timescale separation [23, 72, 156, 200, 201], although there are some other computational approaches exploiting redundancies in the network structure [1], or using *a priori* [8] and *a posteriori* [88, 159] error minimisation strategies. A number of these approaches emphasise the important requirement that model reduction methods for biological systems should result in biologically meaningful state spaces. For example, the $\tilde{N}$ components $\tilde{x}_i$ of the reduced state vector $\tilde{\boldsymbol{x}}$ should ideally correspond to the concentrations of $\tilde{N}$ biomolecular species X_i, thereby modelling biochemically relevant quantities.

1.3 Composition and Decomposition

Model decomposition is another strategy for the analysis, simulation and control of BRN models that is closely related to model reduction. Rather than approximating a large-scale system with a smaller system, a large-scale system can instead be decomposed into a collection of smaller, interconnected subsystems. Much of Systems Theory is concerned with the analysis and design of interconnected systems [217]. A model decomposition strategy provides a hierarchical method that simplifies the analysis of large systems: each component can be analysed in turn before investigating the effect of their interconnection. Indeed, taking this approach with the model reduction problem above results in methods of *structured model reduction* [24, 169, 170, 207]. When applied to the computation of system properties, the hierarchical, modular strategies enabled by model decomposition have the potential to significantly simplify large-scale optimisation tasks.

In addition to Systems Theory, network decomposition also plays an important role in Network Science [27]. Algorithmic methods can be employed to detect the underlying structure of a network in terms of its *community structure*: communities are groups of nodes that are tightly connected, but that have sparse inter-group connections [67, 145]. Apart from community structure, alternative network architectures such as multi-layered networks [106, 132] or core-periphery networks [100] may also be appropriate decomposition frameworks. Numerical optimisation also benefits from strategies based on decomposition; programs can be made significantly faster by an appropriate distribution of the optimisation problem amongst interconnected sub-problems [47, 125].

This thesis will consider model decomposition in the context of analysing large ODE models of BRNs. There are two closely-related biological interpretations of the reformulation of large systems as an interconnection of smaller subsystems. The first of these reflects the purpose of Systems Biology, whereby BRNs are reverse-engineered to understand their functional components and how they interact [30, 34]. The second interpretation arises from Synthetic Biology. The prevailing strategy for the design of large-scale *de novo* biomolecular networks is as an interconnection of well-characterised biomolecular subsystems, known as parts and modules [155, 188]. We will describe each of these biological interpretations in turn.

1.3.1 Systems Biology

As described in Section 1.1, ODE models for BRNs can be built on the scale of an entire cell, but researchers often choose to focus on a particular functional subsystem, such as the glycolytic pathway [37, 41, 94], the heat shock [114] and osmotic shock [71] responses, the chemotaxis pathway [85, 215], the ErbB signalling pathway [44], and so on. Each of these BRNs can be thought of in the context of model decomposition as a subsystem of the larger, cell-level BRN. By focusing on a particular functional subsystem, the complexity of the cell is being overcome not by reducing the dimension of a cellular-scale model, as in the previous section, but instead by considering that subsystem in isolation from its cellular context: there are many more species and reactions simply *not* being modelled. However, when considered in isolation from the rest of the cell, any model of a subsystem is not necessarily accurate. For example, the burden that the subsystem puts on cellular resources [5, 50, 82, 185] may cause unexpected, unmodelled, coupling between apparently unconnected models, significantly altering the conclusions that can be drawn.

The insight given by the model into the behaviour of the system can be increased further by decomposing the system into smaller functional subunits [4, 142, 166, 167]. One approach to network decomposition is to base the definition of subsystems on their system-theoretic roles. For example, the heat shock response can be analysed as an interconnection of subsystems with the roles of plant, actuator, computer, feedback and feedforward sensor [114]. Similarly, chemotaxis can be modelled as the interconnection of modules that each perform specific sensing and actuating tasks [182].

Computationally-driven decomposition techniques are also possible, and the resulting subsystems often have an intuitive *post hoc* justification. For example, methods for the detection of motifs [6, 131] or communities [145] can be used to determine structural components in directed networks formed from the connectivity patterns of, for example, transcription factors in a gene regulatory network [203, 211] or protein–protein interactions [42]. Motifs are small structures that appear in large-scale networks with statistically significant frequency, and which have functional interpretations such as autoregulatory or feedforward loops. It is hypothesised that motifs form a family of simple building blocks of large, complex evolved biomolecular networks [6]. An alternative decomposition technique, in the case of metabolic reaction networks at steady state, is that of elementary flux modes (EFMs) [177, 195, 202]. EFMs decompose the network at steady state into a collection of distinct pathways that reflect the allowed distributions of flux through a homeostatic metabolic network.

Whether subsystems are detected through functional or computational considerations, the difficulty of a decomposed analysis arises in the transition from considering each subsystem in isolation to understanding how they combine their dynamics to give an integrated behaviour. For example, the mechanisms by which biomolecular subsystems interconnect can cause retroactive, or loading, effects [56, 83]. These effects cause the input–output behaviour of a subsystem to change depending on whether it is isolated or interconnected with the rest of the network. Computational methods of decomposition can be designed to minimise retroactivity [165, 186]. Other interaction effects are also important, such as the coupling of processes through competition for shared resources (as mentioned above), or crosstalk between multiple pathways that causes them to respond non-specifically to certain signals [21, 118, 119, 126, 171].

1.3.2 Synthetic Biology

Synthetic Biology is a rapidly-developing field that seeks to exploit the knowledge gained through Systems Biology to engineer new biomolecular circuits and combine them to produce new cellular functions [103, 122]. The relationship between the two fields is analogous to the difference between forward and reverse engineering. The dominant approach to Synthetic Biology rests on the hierarchy of building systems out of interconnected modules (e.g. biomolecular networks or pathways), which are in turn built out of interconnected parts (e.g. smaller pathways or motifs) [188]. Since early successes in building small genetic devices in the early 2000s, such as the toggle switch [70] and an oscillator [62], many more new devices have been designed, such as logic gates [7, 58, 214], other switches [20, 54, 112], oscillators [68, 75, 196], and signalling pathways [187]. The next stage of development for Synthetic Biology is to combine these components into larger systems to achieve a greater degree of functional sophistication [39, 122, 155].

This composition problem is clearly the dual of the decomposition problem in Section 1.3.1. Decomposition identifies the components of a large-scale network for analysis purposes, but in Synthetic Biology the subsystems are clearly defined, and must be designed and tuned so that their interaction in a large-scale synthetic system results in the intended functionality [17]. Nevertheless, the goal of decomposed analysis methods remains the same in Synthetic Biology: to be able to accurately predict how the behaviour of a large-scale BRN will arise from the coordination of its constituent parts.

As described above, interaction effects such as retroactivity and crosstalk significantly influence the behaviour of interconnected subsystems. Subsystems can be designed in order to avoid or control retroactivity [83], or additional subsystems can be inserted to insulate subsystems from its effects [56, 57]. Synthetic devices may be designed and prototyped *in vitro* but then place a burden on the native cellular processes when implemented *in vivo* [185, 199]. In addition to changing how synthetic modules interact with one another, this burden also affects the behaviour of the cellular chassis, the effects of which may propagate through the system to give unexpected and undesired consequences. It is therefore important to be able to predict the extent to which the existing functionality (or viability) of a cellular process can be impaired through the incremental addition of new reactions and species [31, 109]. The problem of reliability becomes even more difficult when variability between cells within a given population is taken into consideration, a difficulty amplified when considering the greater variability between strains, or between species [19, 49, 184]. In this case, the many bi-directional interactions between biosynthetic subsystems and their host cells become extremely uncertain [40]. The effects of these uncertainties need to be carefully managed in order to quantify and robustify the behaviour of future large-scale synthetic systems [147].

In both Systems and Synthetic Biology, the key interpretation of model decomposition is that the large-scale system is made up of smaller components, usually defined as groups of species and reactions and known as *modules*, each of which has a well-defined function and behaviour. However, interaction effects can cause modules to behave differently in different contexts, which may invalidate any correspondences between modules and their function. In this thesis we will show that there are other BRN architectures that are distinct from the commonly-used modular decomposition framework.

1.4 Outline of Thesis Contribution

Recall the discussion of model reduction in Section 1.2. Unlike the balanced truncation of linear input–output systems, reducing nonlinear autonomous systems through singular perturbation does not give *a priori* bounds on the incurred error. The first part of this thesis outlines a computational approach to estimating the worst-case approximation error of any given reduction of an autonomous nonlinear (rational) polynomial model (1.2). This technique relies on a semidefinite programming approach to dissipativity analysis, which is introduced in Chapter 2. In Chapter 3, we show how dissipativity analysis can be applied to the model reduction problem, by providing a means for estimating the difference between two trajectories. We use this technique to propose a model reduction algorithm designed to accept only those reductions that incur approximation errors guaranteed to fall below a particular threshold.

The next contribution of this thesis is in the related area of model decomposition. Section 1.3 identified modular decomposition of biomolecular networks as being the most prevalent method of decomposition. However, as with community detection in static networks, there may be other forms of network architecture that are more suited to BRN models. Chapter 4 introduces a new framework, termed *layering*, for the decomposition of BRNs. One important difference between the layered and modular frameworks is that the contextual effects on modules, which change their input–output behaviour in different contexts, are explicitly accounted for in layered dynamics.

A fundamental question associated with any system decomposition technique is as follows: given a large-scale system to decompose, how should the subsystems be chosen? In Chapter 4, we consider how to ensure the subsystems are as simple as possible. Furthermore, one approach that is particularly applicable to BRNs is to exploit timescale separation. Model reduction by singular perturbation, described in Section 1.2.2, is essentially a *modular* decomposition technique, that seeks to partition the elements of the state vector as fast or slow. Chapter 5 demonstrates that, by adapting singular perturbation to exploit the layered decomposition framework, a model reduction by timescale can occur without requiring transformations of the state into biochemically irrelevant variables.

A fundamental property of any given decomposition of a system into modules or layers is how the behaviour of the isolated subsystems is changed as a result of their interconnection. Chapter 6 seeks to quantify how signals propagate across layered BRNs, focusing in particular on a layered decomposition of the steady state parameter sensitivity analysis. This chapter demonstrates how the difference between each layer's isolated and integrated sensitivities can be used to define a graph that quantifies the effect of interactions between layers. This graph can be used to address the question of choosing a decomposition, by defining an optimal decomposition to have the simplest possible inter-layer communication structure.

Section 1.3 identified decomposition as a strategy for speeding a computation task by distributing it across subsystems. It can be shown that the dissipativity analysis techniques described in Chapter 2 and 3 tend to result in large-scale optimisation problems that are often impractical to solve. Chapter 7 demonstrates the significant computational benefit of applying the layering framework to these problems. The strategy taken is to extend the techniques introduced in Chapter 2 with additional constraints that reflect the propagation of dynamic signals through a layered cascade. Following the approach taken in Chapter 3, this implies that the difficult task of estimating the dynamic response of BRNs to structural and parametric perturbations, including structured model reductions, can be decomposed and thus significantly simplified. Hence, Chapter 7 provides a further illustration of the close relationship between model decomposition and model reduction as complementary techniques for the analysis of large-scale BRNs.

1.4.1 Publications

The work in this thesis has led to the following publications. Reference [150] reports how the semi-definite programming technique for dissipativity analysis, described in Chapter 2, can be applied to the model reduction problem, as illustrated in Chapter 3. In references [151, 152], we introduce the layered framework of Chapter 4 and describe its application to model reduction through timescale separation, as shown in Chapter 5. Reference [153] contains the work of Chapter 4 on the importance of stability in layered decompositions, and also contains the result in Chapter 6 that quantifies how steady-state perturbations propagate between layers. The work of Chapter 7 on structured storage functions has been published as reference [154].

Chapter 2

Dissipativity and SOS Programming

The first part of this thesis is motivated by the model reduction problem introduced in Section 1.2; specifically, the task of constructing an *a priori* estimate of the error incurred by a given model reduction. This preliminary chapter briefly reviews the elements of systems theory and the computational techniques that will subsequently be used in Chapter 3 for this task. First, Section 2.1 introduces the system-theoretic concept of dissipativity and gives a number of examples to illustrate the wide range of system properties that can be formulated using this framework. Second, Section 2.2 reviews a computational technique, called SOS programming, that can be used to relax optimisation problems to make them more amenable to numerical approaches. The rest of the chapter demonstrates the application of SOS programming to certifying the dissipativity of a particular class of uncertain nonlinear systems, which are subject to constraints on state and input values. The SOS approach to dissipativity analysis will form the basis of the contribution of Chapter 3 to the model reduction error estimation problem.

2.1 Dissipativity

Dissipativity is a fundamental property of any ODE model of a physical system [84, 111, 212, 213]. Consider the dynamics

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big), \tag{2.1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}\big(\boldsymbol{x}(t)\big), \tag{2.1b}$$

where the state $\boldsymbol{x}$ takes values $\boldsymbol{x}(t) \in \mathscr{D}$ in the domain $\mathscr{D} \subseteq \mathbb{R}^N$, the input $\boldsymbol{u}$ takes values $\boldsymbol{u}(t) \in \mathscr{U}$ for the set of input values $\mathscr{U} \subseteq \mathbb{R}^p$, and the output $\boldsymbol{y}$ takes values $\boldsymbol{y}(t) \in \mathbb{R}^q$. Assume also that the system has a set of allowed initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathscr{X}_0$ for a given subset of the domain $\mathscr{X}_0 \subseteq \mathscr{D}$.

Dissipativity is a property of the ODE system (2.1) that describes, for a generalised concept of 'energy', how the energy stored internally to the system relates to an external energy supply. At each time point $t \geq 0$, the internally stored energy is quantified by a *storage function* $V\big(\boldsymbol{x}(t)\big)$, which is a function of the internal state $\boldsymbol{x}$, while the external *supply rate* is a function of the input–output signals $s\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big)$. The dissipativity of (2.1) with respect to the given supply rate is defined as follows.

Definition 2.1 (Adapted from [84]) The system (2.1) is *dissipative with respect to the supply rate* $s : \mathbb{R}^{p+q} \to \mathbb{R}$ if there exists a function $V : \mathscr{D} \to \mathbb{R}$ such that $V(\boldsymbol{x}) \geq 0$ and

$$\dot{V}\big(\boldsymbol{x}(t)\big) = \nabla V\big(\boldsymbol{x}(t)\big) \cdot \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \leq s\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big) \tag{2.2}$$

for state and input values $\boldsymbol{x}(t) \in \mathscr{D}$ and $\boldsymbol{u}(t) \in \mathscr{U}$. The function V is called a *storage function* of the system with respect to s. ♢

An alternative condition to (2.2), which avoids the assumption that V is differentiable with respect to $\boldsymbol{x}$, is given by

$$V\big(\boldsymbol{x}(T_2)\big) \leq V\big(\boldsymbol{x}(T_1)\big) + \int_{T_1}^{T_2} s\big(\boldsymbol{u}(t), \boldsymbol{y}(t)\big)\, \mathrm{d}t \tag{2.3}$$

for $0 \leq T_1 \leq T_2$. This alternative expression emphasises the physical intuition of dissipativity: the change in internal energy V over the time interval $[T_1, T_2]$ is bounded above by the total external energy supplied, as measured by $s(\boldsymbol{u}, \boldsymbol{y})$. However, the storage functions that will be computed in this thesis will be polynomial, and therefore differentiable, in $\boldsymbol{x}$. Hence, the derivative condition (2.2) on V will be used to certify dissipativity.

The dissipativity framework has been used to analyse a number of different system properties, each of which can be characterised by the appropriate choice of supply rate [60].

Example 2.2 (Passivity, [111]) Consider the system (2.1) with equal input and output dimensions $p = q$. Suppose that this system is dissipative with respect to the supply rate $s(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u}^T \boldsymbol{y}$. This property is known as the *passivity* of the system. By this definition of s, the supply rate is interpreted as a measure of the net rate at which energy is being transferred into the system from its environment. An alternative interpretation of the supply rate is that its negative value $-s$ defines the net rate at which energy is extracted from the system into its environment. The dissipative system satisfies (2.3), implying that the net energy transferred into the system over any time interval is an upper bound on the change in internal storage. Similarly, the net amount of energy that can be extracted from the system in any given time interval is, at most, the internal energy stored at the start of the interval. We can deduce that the system does not generate its own internal energy, hence the term *passive.* ♢

Example 2.3 (Induced $\mathscr{L}_2 \to \mathscr{L}_2$ gain) Suppose that the system (2.1), now with general input and output dimensions p and q, is dissipative with respect to the supply rate $s(\boldsymbol{u}, \boldsymbol{y}) = \gamma^2 |\boldsymbol{u}|^2 - |\boldsymbol{y}|^2$ for a given constant $\gamma \in \mathbb{R}$. Substituting this supply rate into (2.3) implies that the energy $\|\boldsymbol{y}\|^2$ of the output $\boldsymbol{y}$, over any given time interval, is at most a multiple $\gamma^2 \|\boldsymbol{u}\|^2$ of the input energy, plus the initial internal storage lost over that interval. If there is no internal storage at the start of the interval, then we can conclude that the induced $\mathscr{L}_2$ to $\mathscr{L}_2$ gain of the system is bounded above by $|\gamma|$. ♢

Example 2.4 (Maximal output energy) Consider the autonomous system (2.1), where the input $\boldsymbol{u}$ has been set to zero. Suppose that this system is dissipative with respect to the supply rate $s(\boldsymbol{u}, \boldsymbol{y}) = -|\boldsymbol{y}|^2$. The inequality (2.3) implies that the output energy $\|\boldsymbol{y}\|^2$, evaluated over a given time interval, is bounded above by the loss in internal storage in that time interval. Since

$V\big(\boldsymbol{x}(t)\big) \geq 0$, this means that the maximum amount of energy that can be observed in the autonomous system is bounded above by the initial storage $V(\boldsymbol{x}_0)$.

Given that the initial conditions are assumed to belong to a space $\mathscr{X}_0 \subseteq \mathscr{D} \subseteq \mathbb{R}^N$ of possible initial conditions, we can conclude that the largest output energy possible, given this set of initial conditions, is at most $\max_{\boldsymbol{x}_0 \in \mathscr{X}_0} V(\boldsymbol{x}_0)$. Alternatively, suppose that we wish to *design* the set $\mathscr{X}_0$ of initial conditions to ensure that the output energy $\|\boldsymbol{y}\|^2$ is no larger than a specific value Y^2. In this case, the sub-level sets of V can be used to constrain $\mathscr{X}_0$ by enforcing $\mathscr{X}_0 \subseteq \big\{\boldsymbol{x}_0 \in \mathscr{D} \mid V(\boldsymbol{x}_0) \leq Y^2\big\}$. ◊

Example 2.5 (The reachable subspace) A dual property to the bounds on the output energy above is found by allowing $\boldsymbol{u}$ to be non-zero, and certifying the dissipativity of (2.1) with respect to the supply rate $s(\boldsymbol{u}, \boldsymbol{y}) = |\boldsymbol{u}|^2$. The relationship (2.3) implies that the increase in internally stored energy over a given time interval is bounded above by the energy $\|\boldsymbol{u}\|^2$ of the input applied during that interval. Here it will be assumed that the initial condition $\boldsymbol{x}_0 = \boldsymbol{0}$ is fixed, and that $V(\boldsymbol{0}) = 0$.

Suppose that there is a constraint $\|\boldsymbol{u}\|^2 \leq U^2$ on the maximum possible energy of the input over a given time interval $[0, T]$. Dissipativity implies that U^2 is an upper bound on the value of $V\big(\boldsymbol{x}(T)\big)$. Consider the *reachable set* $\mathscr{R}(T) \subseteq \mathscr{D}$ of possible values of $x(T)$. This set can be estimated by the sub-level set $\mathscr{R}(T) \subseteq \big\{\boldsymbol{x} \in \mathscr{D} \mid V(\boldsymbol{x}) \leq U^2\big\}$. Conversely, given a desired target set of states $\mathscr{T}(T) \subset \mathscr{D}$, the smallest energy of the input applied over $[0, T]$ required to bring the system to $\mathscr{T}(T)$ is at least $\min_{\boldsymbol{x} \in \mathscr{T}(T)} V(\boldsymbol{x})$. ◊

The examples above illustrate a number of system properties that can be formulated in terms of dissipativity. The first two problems demonstrate that the certification of dissipativity is essentially a feasibility problem, where it is required that there exists a feasible storage function V that satisfies the constraint (2.2). The final two examples demonstrate that, depending on the property being measured, this problem can be extended into an optimisation problem by defining a particular objective functional of V.

In particular, consider the task in Example 2.4 of finding the *optimal* storage function V for a given fixed set $\mathscr{X}_0$ of possible initial conditions, such that $\max_{\boldsymbol{x}_0 \in \mathscr{X}_0} V(\boldsymbol{x}_0)$ is minimal. This provides the tightest upper bound on the largest possible output energy $\|\boldsymbol{y}\|^2$. Alternatively, when designing $\mathscr{X}_0$, the optimisation task is to provide the largest set of initial conditions $\boldsymbol{x}_0 \in \mathscr{X}_0$ while still ensuring $\|\boldsymbol{y}\|^2 \leq Y^2$, by finding a feasible storage function with a maximal sub-level set volume. Similarly, Example 2.5 also suggests two possible optimisation problems: finding the best lower bound on the input energy needed to bring the system state to $\mathscr{T}(T)$; or finding a best estimate of the reachable set $\mathscr{R}(T)$ given a maximum input energy U^2.

In general, the formulation of dissipativity certification as an optimisation problem is as follows.

Program 2.6 Find a storage function $V : \mathscr{D} \to \mathscr{R}$ satisfying

$$\min_V P(V)$$

$$\text{s.t.} \qquad V(\boldsymbol{x}) \geq 0 \tag{2.4a}$$

$$s(\boldsymbol{u}, \boldsymbol{y}) - \nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \geq 0 \tag{2.4b}$$

for all vectors $\boldsymbol{x} \in \mathscr{D}$ and $\boldsymbol{u} \in \mathscr{U}$. Here $P(V) \in \mathbb{R}$ is an objective functional; by setting $P(V) = 0$, this program can be reduced to testing feasibility. ◊

2.2 Sum of Squares Programming

In the remainder of this chapter we introduce a technique designed to relax the positive semi-definite constraints (2.4) for a particular class of vector fields $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, and also to provide convex estimates of the state and input constraints $\boldsymbol{x}(t) \in \mathscr{D}$ and $\boldsymbol{u}(t) \in \mathscr{U}$. This section reviews the Sum of Squares (SOS) method for the relaxation of positive semi-definite constraints on polynomials.

2.2.1 Sum of Squares polynomials

Consider the space $\mathbb{R}[\boldsymbol{x}]$ of polynomials $p : \mathbb{R}^N \to \mathbb{R}$, where each $p(\boldsymbol{x})$ is a finite linear combination of monomials $x_1^{k_1} x_2^{k_2} \dots x_N^{k_N}$ for non-negative integers $k_i \in \mathbb{Z}_{\geq 0}$. Define also the set $\mathbb{R}[\boldsymbol{x}]_{\geq 0} \subseteq \mathbb{R}[\boldsymbol{x}]$ of positive semi-definite polynomials such that $p(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$. In general, testing whether $p \in \mathbb{R}[\boldsymbol{x}]_{\geq 0}$ is NP-hard [143]. However, the following results imply that there is a stronger condition on p which is not as difficult to test, but which guarantees that $p \in \mathbb{R}[\boldsymbol{x}]_{\geq 0}$.

Definition 2.7 Suppose that, given $p \in \mathbb{R}[\boldsymbol{x}]$, there exist k polynomials $p_1, \dots, p_k \in \mathbb{R}[\boldsymbol{x}]$ such that p can be written

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} \big(p_i(\boldsymbol{x})\big)^2.$$

This property is called "p is Sum of Squares (SOS)". The set of all SOS polynomials is denoted by $\Sigma[\boldsymbol{x}] = \big\{p \in \mathbb{R}[\boldsymbol{x}] \mid p \text{ is SOS}\big\}$. ◊

Proposition 2.8. *The set $\Sigma[\boldsymbol{x}]$ of SOS polynomials is a strict subset of the set of positive semi-definite polynomials $\mathbb{R}[\boldsymbol{x}]_{\geq 0}$.*

Proof. It is clear that $p \in \Sigma[\boldsymbol{x}]$ implies that $p(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$. To show the strictness of the inclusion, it suffices to find an example of $p \in \mathbb{R}[\boldsymbol{x}]_{\geq 0} \setminus \Sigma[\boldsymbol{x}]$. It can be shown that the Motzkin form $M(x, y, z) = x^4 y^2 + x^2 y^4 + z^6 - 3x^2 y^2 z^2$ is an example of a positive semi-definite polynomial that is not SOS [143]. □

Lemma 2.9. *The SOS condition $p \in \Sigma[\boldsymbol{x}]$ is equivalent to the existence of a positive semi-definite constant matrix $\boldsymbol{Q}$ and vector $\boldsymbol{z}(\boldsymbol{x})$ of monomials of $\boldsymbol{x}$ such that $p(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{Q} \boldsymbol{z}(\boldsymbol{x})$.*

Proof. Define the polynomial $p(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{Q} \boldsymbol{z}(\boldsymbol{x})$ for positive semi-definite $\boldsymbol{Q}$. Consider the Cholesky decomposition of $\boldsymbol{Q} = \boldsymbol{L}^T \boldsymbol{L}$. Then $p(\boldsymbol{x}) = (\boldsymbol{L}\boldsymbol{z}(\boldsymbol{x}))^T (\boldsymbol{L}\boldsymbol{z}(\boldsymbol{x}))$ is a SOS polynomial, where each $p_i(\boldsymbol{x}) = \boldsymbol{L}_i \boldsymbol{z}(\boldsymbol{x})$ for each row $\boldsymbol{L}_i$ of $\boldsymbol{L}$. Conversely, suppose $p(\boldsymbol{x}) = \sum (p_i(\boldsymbol{x}))^2$. Construct the vector $\boldsymbol{z}(\boldsymbol{x})$ of all of the monomials used in at least one p_i. Then each polynomial p_i can be written $\boldsymbol{L}_i \boldsymbol{z}(\boldsymbol{x})$ for row vector $\boldsymbol{L}_i$ of the coefficients of each of the monomials. By forming a matrix $\boldsymbol{L}$, the rows of which are equal to the row vectors $\boldsymbol{L}_i$ defining each p_i, we can therefore re-write $p(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{Q} \boldsymbol{z}(\boldsymbol{x})$ for the trivially positive semi-definite matrix $\boldsymbol{Q} = \boldsymbol{L}^T \boldsymbol{L}$. □

Lemma 2.9 implies that testing $p \in \Sigma[\boldsymbol{x}]$ is equivalent to testing the existence of a positive semi-definite matrix $\boldsymbol{Q}$ such that $p = \boldsymbol{z}^T \boldsymbol{Q} \boldsymbol{z}$. This is a convex feasibility problem that can be solved in polynomial time [143], and is therefore a relaxation of the problem of testing $p \in \mathbb{R}[\boldsymbol{x}]_{\geq 0}$. However, Proposition 2.8 is proof that $p \in \Sigma[\boldsymbol{x}]$ is a stronger condition. There has been a great deal of interest in the gap between $\Sigma[\boldsymbol{x}]$ and $\mathbb{R}[\boldsymbol{x}]_{\geq 0}$ [45, 46], which is the cost of making the problem computationally tractable through semi-definite programming [29]. Furthermore, recent results have identified smaller subsets of $\mathbb{R}[\boldsymbol{x}]_{\geq 0}$ that will require much cheaper linear or second-order cone programming in order to test set membership [2].

2.2.2 Convex optimisation in SOSTOOLS

The SOS approach can be applied to feasibility and optimisation problems by relaxing positive semi-definite constraints into SOS constraints. Consider Program 2.6, where we take $P(V) = 0$ to make this a feasibility problem. Suppose also that there are no constraints on the set of state values $\mathscr{D} = \mathbb{R}^N$ and input values $\mathscr{U} = \mathbb{R}^p$. The SOS relaxation of the positive semi-definite constraints (2.4) results in the following example of an SOS program.

Program 2.10 Find a polynomial $V \in \mathbb{R}[\boldsymbol{x}]$ subject to the SOS constraints

$$V(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{2.5a}$$

$$s(\boldsymbol{u}, \boldsymbol{y}) - \nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \in \Sigma[\boldsymbol{x}], \tag{2.5b}$$

for all $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{u} \in \mathbb{R}^p$. ◊

The solutions to SOS programs such as these can be computed using specifically-developed software. The toolbox that will be used in this thesis is SOSTOOLS [139], which is an implementation of SOS programming in MATLAB that enables optimisation programs to be defined with SOS constraints. In Program 2.10, the SOS decision variable V satisfying (2.5a) can be defined through SOSTOOLS by the function `V = sossosvar(p,z)` for the previously-declared SOS program name `p` and a vector of monomials `z`. Similarly, the constraint (2.5b) can be implemented by the SOS inequality `p = sosineq(p,s-Vdot)`. Each of these functions converts the polynomial SOS constraint into the equivalent constraint on positive semi-definite $\boldsymbol{Q}$ according to Lemma 2.9. Thus SOSTOOLS translates the SOS program into a semi-definite program that can be solved by any of a number of solvers, such as SeDuMi [197]. The solution of the semi-definite program is then returned to SOSTOOLS, and converted back into polynomial form. For a more complete example of the use of SOSTOOLS to solve a SOS program, see Appendix A.

2.3 Extensions of the SOS Dissipativity Program

The basic SOS program for certifying dissipativity is given by Program 2.10, but is not widely applicable in this form. In particular, V is being sought to satisfy (2.5b) over all $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{u} \in \mathbb{R}^p$, and the system may not be dissipative for all such values. However, the original ODEs (2.1) have been defined for constrained state and input values $\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \in \mathscr{D} \times \mathscr{U}$, for which it may be possible to prove dissipativity. Furthermore, if at least one of the vector field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ or output map $\boldsymbol{h}(\boldsymbol{x})$ is not polynomial, then the SOS condition (2.5b) cannot hold. Finally, depending on the application, parametric uncertainty may be an important feature, where dissipativity should be certified for a range of possible parameter values. This section considers the extensions to Program 2.10 that will overcome each of these problems [140, 141, 143].

2.3.1 Non-global storage functions

The original dissipativity program given by Program 2.6 also includes constraints on the state values $\boldsymbol{x}(t) \in \mathscr{D}$ and input values $\boldsymbol{u}(t) \in \mathscr{U}$. A generalisation of the S-procedure [140, 143] can be used to relax the condition (2.5b) accordingly.

Suppose that there exists an estimate of $\mathscr{D} \times \mathscr{U}$ given by the semi-algebraic set

$$\mathscr{D} \times \mathscr{U} \subseteq \left\{(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^{N+p} \mid \phi_i(\boldsymbol{x}, \boldsymbol{u}) \le 0 \text{ for all } i = 1, \dots, R\right\} = \Phi,$$

for polynomial functions $\phi_i \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{u}]$. If (2.4b) can be shown to hold for all $(\boldsymbol{x}, \boldsymbol{u}) \in \Phi$, then it follows that it must be true for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$, and thus dissipativity is certified for the given sets of state values $\mathscr{D}$ and input values $\mathscr{U}$. This task can be implemented in the following relaxation of Program 2.10.

Program 2.11 Find the $R+1$ polynomials $V \in \mathbb{R}[\boldsymbol{x}]$ and $\sigma_i \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{u}]$, for $i = 1, \dots, R$, subject to the SOS constraints

$$V(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{2.6a}$$

$$\sigma_i(\boldsymbol{x}, \boldsymbol{u}) \in \Sigma[\boldsymbol{x}, \boldsymbol{u}], \tag{2.6b}$$

$$\left[s(\boldsymbol{u}, \boldsymbol{y}) - \nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})\right] + \sum_{i=1}^{R} \sigma_i(\boldsymbol{x}, \boldsymbol{u}) \phi_i(\boldsymbol{x}, \boldsymbol{u}) \in \Sigma[\boldsymbol{x}, \boldsymbol{u}] \tag{2.6c}$$

for the polynomials ϕ_i defining Φ. ◊

Proposition 2.12. *Assume that the vector field $\boldsymbol{f}$ of the dynamics* (2.1) *is polynomial. If feasible polynomials V and σ_i can be found for the SOS Program 2.11, then V is a storage function certifying the dissipativity of* (2.1) *with respect to $s(\boldsymbol{u}, \boldsymbol{y})$, for the constrained state values $\boldsymbol{x}(t) \in \mathscr{D}$ and input values $\boldsymbol{u}(t) \in \mathscr{U}$.*

Proof. Suppose feasible SOS polynomials $V \in \Sigma[\boldsymbol{x}]$ and $\sigma_i \in \Sigma[\boldsymbol{x}, \boldsymbol{u}]$ exist. Since each σ_i is positive semi-definite, the estimate $\mathscr{D} \times \mathscr{U} \subseteq \Phi$ implies that

$$\sum_{i=1}^{R} \sigma_i(\boldsymbol{x}, \boldsymbol{u}) \phi_i(\boldsymbol{x}, \boldsymbol{u}) \le 0$$

for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$. This inequality, together with condition (2.6c), implies that $s(\boldsymbol{u}, \boldsymbol{y}) - \dot{V} \ge 0$ along the solutions to (2.1). Hence (2.2) is satisfied, and dissipativity is certified. □

2.3.2 Non-polynomial vector fields

We can extend the use of SOS techniques to a larger class of systems than those with a polynomial vector field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$. Instead, suppose $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ is a rational polynomial, such that each component f_i for $i = 1, \dots, N$ can be written

$$f_i(\boldsymbol{x}, \boldsymbol{u}) = \frac{g_i(\boldsymbol{x}, \boldsymbol{u})}{\psi(\boldsymbol{x}, \boldsymbol{u})}$$

for the polynomials $g_i \in \mathbb{R}[\boldsymbol{x},\boldsymbol{u}]$ and polynomial common denominator $\psi \in \mathbb{R}[\boldsymbol{x},\boldsymbol{u}]$. We assume that $\boldsymbol{f}$ is well-defined, which means that ψ cannot take the value 0. Since ψ is a polynomial, it is continuous, and this together with $\psi \neq 0$ implies that ψ cannot change sign in $\mathscr{D} \times \mathscr{U}$. Without loss of generality, we assume that $\psi(\boldsymbol{x},\boldsymbol{u}) > 0$ for all $(\boldsymbol{x},\boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$.

Clearly neither of the dissipativity conditions (2.5b) or its local variant (2.6c) can hold, because $\nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$ is not a polynomial. Assume in what follows that dissipativity is required to hold only for $(\boldsymbol{x},\boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$.

Program 2.13 Find the $R+1$ polynomials $V \in \mathbb{R}[\boldsymbol{x}]$ and $\sigma_i \in \mathbb{R}[\boldsymbol{x},\boldsymbol{u}]$, for $i = 1,\dots,R$, subject to the SOS constraints

$$V(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \tag{2.7a}$$

$$\sigma_i(\boldsymbol{x},\boldsymbol{u}) \in \Sigma[\boldsymbol{x},\boldsymbol{u}] \tag{2.7b}$$

$$\psi(\boldsymbol{x},\boldsymbol{u})\left[s(\boldsymbol{u},\boldsymbol{y}) - \nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})\right] + \sum_{i=1}^{R} \sigma_i(\boldsymbol{x},\boldsymbol{u})\phi_i(\boldsymbol{x},\boldsymbol{u}) \in \Sigma[\boldsymbol{x},\boldsymbol{u}]. \tag{2.7c}$$

◊

Proposition 2.14. *If the SOS Program 2.13 is feasible with storage function V, then V certifies the dissipativity of the rational polynomial system* (2.1) *with respect to $s(\boldsymbol{u},\boldsymbol{y})$.*

Proof. As in the proof of Proposition 2.12, the summation term $\sum \sigma_i \phi_i$ in (2.7c) is non-positive for $(\boldsymbol{x},\boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$, and therefore $\psi(s - \dot{V}) \geq 0$. Hence, since we have assumed that $\psi > 0$ for all $(\boldsymbol{x},\boldsymbol{u}) \in \mathscr{D} \times \mathscr{U}$, this inequality implies (2.2). □

2.3.3 Uncertain parameters

Now suppose also that the dynamics (2.1) depend on r uncertain constrained parameter values $\boldsymbol{\pi} \in \mathscr{P} \subseteq \mathbb{R}^r$, where the dependence is denoted by the vector field $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u};\boldsymbol{\pi})$. Recall the estimate of the state and input spaces $\mathscr{D}\times\mathscr{U}$ with Φ. This approach can be extended to include the parameter space $\mathscr{P}$ by defining an estimate through the semi-algebraic set

$$\mathscr{D} \times \mathscr{U} \times \mathscr{P} \subseteq \left\{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}) \in \mathbb{R}^{N+p+r} \mid \phi_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}) \leq 0 \text{ for all } i = 1,\dots,R\right\} = \Phi$$

for the R polynomial functions $\phi_i \in \mathbb{R}[\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}]$. The following SOS program provides for constrained parameter uncertainties by introducing additional auxiliary variables $\boldsymbol{\pi}$ to the variables $\boldsymbol{x}$ and $\boldsymbol{u}$. Note that the storage function $V \in \mathbb{R}[\boldsymbol{x}]$ is independent of $\boldsymbol{\pi}$, thereby certifying dissipativity for all possible values of $\boldsymbol{\pi} \in \mathscr{P}$.

Program 2.15 Find the $R+1$ polynomials $V \in \mathbb{R}[\boldsymbol{x}]$ and $\sigma_i \in \mathbb{R}[\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}]$, for $i = 1,\dots,R$, subject to the SOS constraints

$$V(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{2.8a}$$

$$\sigma_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}) \in \Sigma[\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}], \tag{2.8b}$$

$$\begin{aligned}\psi(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi})&\left[s(\boldsymbol{u},\boldsymbol{y}) - \nabla V(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u};\boldsymbol{\pi})\right] \\ &+ \sum_{i=1}^{R} \sigma_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi})\phi_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}) \in \Sigma[\boldsymbol{x},\boldsymbol{u},\boldsymbol{\pi}].\end{aligned} \tag{2.8c}$$

$\diamondsuit$

Proposition 2.16. *If the SOS Program 2.15 is feasible with storage function V, then V certifies the dissipativity of the uncertain rational polynomial system with respect to $s(\boldsymbol{u}, \boldsymbol{y})$.*

Proof. See the proof of Proposition 2.14, with additional $\boldsymbol{\pi}$ dependence. □

Note that V is an SOS polynomial in $\boldsymbol{x}$ only. Thus, for any value of $\boldsymbol{\pi} \in \mathscr{P}$, the dissipativity inequality (2.3) holds; although the integral term varies with $\boldsymbol{\pi}$, the storage function is independent of $\boldsymbol{\pi}$. A possible generalisation of Program 2.15 would be to adapt (2.8a) so that $V(\boldsymbol{x}, \boldsymbol{\pi}) \in \Sigma[\boldsymbol{x}, \boldsymbol{\pi}]$ is no longer independent of $\boldsymbol{\pi}$. Nevertheless, Program 2.15 is the most general SOS program for the certification of dissipativity that will be considered in this thesis. It can be used to certify the dissipativity with respect to a given $s(\boldsymbol{u}, \boldsymbol{y})$ of (2.1) for the rational polynomial vector field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\pi})$ with state, input and parameter values $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}) \in \mathscr{D} \times \mathscr{U} \times \mathscr{P}$.

2.3.4 Objective functions

As demonstrated by the examples in Section 2.1, any of the feasibility programs above that implement the certification of dissipativity may be extended to an optimisation by defining an objective functional of the storage function V. Consider again Example 2.4, where $s(\boldsymbol{u}, \boldsymbol{y}) = -\boldsymbol{y}^T \boldsymbol{y}$. Depending on the context of the optimisation, the goal is either to minimise $\max_{\boldsymbol{x}_0 \in \mathscr{X}_0} V(\boldsymbol{x}_0)$ or maximise the volume of the level set $\left\{\boldsymbol{x} \in \mathscr{D} \mid V(\boldsymbol{x}) \leq Y^2\right\}$. Program 2.10 and its extensions above can be extended with an objective functional as follows.

Suppose first that the set of possible initial conditions $\mathscr{X}_0 = \{\boldsymbol{x}_0\}$ is a fixed single point. The objective of the dissipativity analysis in this case would be to minimise $V(\boldsymbol{x}_0)$. However, if $\mathscr{X}_0$ is larger than a singleton set, then minimising $V(\boldsymbol{x}_0)$ for a single $\boldsymbol{x}_0 \in \mathscr{X}_0$ is in general a suboptimal approach. One heuristic that has been applied in this case [22, 150, 170] is to minimise the convex objective function $\operatorname{trace}(\boldsymbol{Q}_V)$, where $\boldsymbol{Q}_V$ is the matrix of decision variables defining the SOS representation of $V(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})^T \boldsymbol{Q}_V \boldsymbol{z}(\boldsymbol{x})$. In the case where $\boldsymbol{z}(\boldsymbol{x}) = \boldsymbol{x}$, minimising $\operatorname{trace}(\boldsymbol{Q}_V)$ is equivalent to maximising the volume of the elliptic level sets of V. This minimisation can also be interpreted as having the effect of slowing the growth of V with $|\boldsymbol{x}|$, and might therefore be a reasonable heuristic for minimising the maximum value of V in $\mathscr{X}_0$. However, in the next chapter, we will show that this heuristic can be severely suboptimal, and demonstrate a novel approach that further exploits the SOS inequality relaxation to give an improved worst-case error bound.

2.4 Computational Cost

Consider the SOS program for certifying the dissipativity of uncertain rational polynomial systems, given by Program 2.15. The $R+1$ SOS polynomials V and σ_i satisfying (2.8a) and (2.8b) define the decision variables of the semi-definite program constructed by SOSTOOLS. In particular, each of the decision polynomials V and σ_i has an associated matrix $\boldsymbol{Q}_V$ or $\boldsymbol{Q}_{\sigma_i}$, each of which will be constrained to be positive semi-definite. The number of decision variables contributed by each polynomial to the semi-definite formulation of Program 2.15 is simply the number of elements of $\boldsymbol{Q}_V$ or $\boldsymbol{Q}_{\sigma_i}$ [139]. Table 2.1 records the number of elements of the matrix $\boldsymbol{Q}_s$ for various variable dimensions and degrees of a generic SOS polynomial $s \in \Sigma[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}]$. Clearly the size

$N+p+r$	$\deg s = 2$	$\deg s = 4$	$\deg s = 6$
2	4	25	81
4	16	196	1,156
6	36	729	6,889
8	64	1,936	26,896
10	100	4,225	81,225
12	144	8,100	206,116
14	196	14,161	461,041

Table 2.1: Growth of number of decision variables associated with $s \in \Sigma[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}]$ with increasing $\deg s$ and increasing dimension of $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}) \in \mathbb{R}^{N+p+r}$.

of the resulting SOS program grows extremely rapidly with both variable dimension and polynomial degree, and can quickly render the SOS program computationally intractable for high-dimensional systems with many inputs and many uncertain parameters.

Another source of computational cost is the dissipativity constraint (2.8c). This condition defines a further matrix $\boldsymbol{Q}$ that is also constrained to be positive semi-definite. The size of $\boldsymbol{Q}$, and hence the number of additional decision variables, is also given by Table 2.1. If the degree of the polynomial in (2.8c) is high then this can contribute an unmanageably large number of additional decision variables. For example, the common denominator $\psi \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}]$ or the polynomials $\phi_i \in \mathbb{R}[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}]$ used to construct the set estimate Φ may be of relatively high degree. These can therefore cause the SOSTOOLS approach to certifying dissipativity to become intractable even with relatively small values of $N + p + r$.

Finally, consider again the R SOS multipliers $\sigma_i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}) \in \Sigma[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\pi}]$. The entries in Table 2.1 correspond to the computational cost contributed by *each* multiplier. We can assume that the number R of ϕ_i used in the construction of Φ is approximately equal to the dimension of $\mathscr{D} \times \mathscr{U} \times \mathscr{P}$, such that $R \approx N + p + r$. For example, the state space $\mathscr{D}$ may be constrained to lie in the rectangular region defined by $\phi_j = (x_j - \underline{x}_j)(x_j - \bar{x}_j)$ for lower and upper bounds $\underline{x}_j$ and $\bar{x}_j$ on each variable x_j, and similarly for $\mathscr{U}$ and $\mathscr{P}$. Then, as the dimension $N + p + r$ increases, the computational cost contributed by each σ_i increases, but so too does the total number R of multipliers. Hence, the number of decision variables increases even faster with N, p, or r than indicated by Table 2.1.

Techniques that exploit the properties of a polynomial's structure, such as sparsity or multipartite structure, in order to reduce the large size of the resulting SOS semi-definite problems have been implemented in SOSTOOLS [139, 143]. Note also that the approximation Φ for $\mathscr{D} \times \mathscr{U} \times \mathscr{P}$ can be formulated in a number of ways, by different choices of ϕ_i. Indeed, an SOS optimisation approach is one method for the tight approximation of given sets [53]. Different constructions of Φ lead to trade-offs between the tightness of the approximation of $\mathscr{D} \times \mathscr{U} \times \mathscr{P}$, the number R of polynomials (and therefore the number of multipliers σ_i), and the degrees of ϕ_i. Careful use of these degrees of freedom may allow for significant computational savings.

Notwithstanding the importance of the degrees of each g_i, ϕ_i, and ψ, a fundamental obstacle to the application of SOS programming methods as a computational tool for dissipativity analysis is the rapid increase in the number of decision variables as the total number of system dimensions, input channels and uncertain parameters $N + p + r$ grows. As described in Section 1.3, a key motivation for the decomposition of large-scale systems is to make the analysis of system properties easier [87, 98, 128, 129, 169]. In Chapter 7, we will adapt the theory described in this chapter to decompose the SOS dissipativity programs described in Section 2.3 into multiple, smaller programs, by seeking storage functions that reflect the particular structure of a large-scale system.

This chapter has briefly discussed recently established results that provide computational tools for dissipativity analysis. The following chapter will describe how these techniques can be used to compare the output trajectories of two dynamical systems. Thus, the trajectories of a large-scale model and its lower-dimensional approximation can be compared through semidefinite programming.

Chapter 3

Trajectory Comparison: Error Bounds

Reducing the dimension of an ODE model usually incurs error in the output trajectory; the best possible model reduction should incur least error across all possible initial conditions and inputs. This chapter applies the tools of SOS programming and dissipativity analysis, introduced in Chapter 2, to construct guaranteed *a priori* upper bounds on the difference between the output trajectories of two autonomous dynamical systems across all allowed initial conditions. By comparing the outputs of a high dimensional model and a reduced model of lower dimension, this technique can be used to estimate the error incurred through the structured model order reduction of BRNs [150]. We will demonstrate the use of this technique for estimating the reduction error by applying it to a singularly perturbed model of enzyme kinetics. The comparison technique introduced in this chapter has a range of additional applications, allowing the effects of parametric perturbations, parametric uncertainty, or retroactivity to be quantified. A second example will illustrate the use of this technique to bound the effect of parametric perturbations of an uncertain system. The chapter will conclude with a preliminary description of how the computational estimation of reduction error can enable an algorithmic approach to systematic model order reduction, and a discussion of how to minimise the conservatism and computational burden of this approach.

3.1 Error Estimation

This chapter is primarily motivated by the model order reduction problem described in Section 1.2. The state $\boldsymbol{x}$ of an autonomous model Γ takes values $\boldsymbol{x}(t) \in \mathscr{D}_\Gamma \subseteq \mathbb{R}^N$ in a state space $\mathscr{D}_\Gamma$ of dimension N, while the state $\tilde{\boldsymbol{x}}$ of the reduced model $\tilde{\Gamma}$ takes values $\tilde{\boldsymbol{x}}(t) \in \mathscr{D}_{\tilde{\Gamma}} \subseteq \mathbb{R}^{\tilde{N}}$ in a state space $\mathscr{D}_{\tilde{\Gamma}}$ of reduced dimension $\tilde{N} < N$. We define the error incurred by the reduction of Γ to $\tilde{\Gamma}$ as the largest value over all possible initial conditions $\boldsymbol{x}_0$ of the functional $\Delta\big(\Gamma(\boldsymbol{x}_0), \tilde{\Gamma}(\tilde{\boldsymbol{x}}_0)\big)$ measuring the difference between the respective output trajectories $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$. The model reduction problem is to select the reduced model so that the largest value of Δ over all of the initial conditions $\boldsymbol{x}_0$ and $\tilde{\boldsymbol{x}}_0 = \chi(\boldsymbol{x}_0)$ is as small as possible.

This is a specific example of a more generic problem. Consider the two autonomous dynamical systems

$$\text{Nominal model, }\Gamma\quad\begin{cases}\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{\pi}),\\ \boldsymbol{y}=\boldsymbol{h}(\boldsymbol{x}),\\ \boldsymbol{x}(0)=\boldsymbol{x}_0;\end{cases}\qquad \text{Perturbed model, }\tilde{\Gamma}\quad\begin{cases}\dot{\tilde{\boldsymbol{x}}}=\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\pi}}),\\ \tilde{\boldsymbol{y}}=\tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}}),\\ \tilde{\boldsymbol{x}}(0)=\tilde{\boldsymbol{x}}_0=\boldsymbol{\chi}(\boldsymbol{x}_0),\end{cases}\tag{3.1}$$

for $\boldsymbol{x}\in\mathscr{D}_\Gamma\subseteq\mathbb{R}^N_{\geq 0}$ and $\tilde{\boldsymbol{x}}\in\mathscr{D}_{\tilde{\Gamma}}\subseteq\mathbb{R}^{\tilde{N}}_{\geq 0}$, where $\tilde{\cdot}$ denotes a perturbed model. The initial condition $\boldsymbol{x}_0$ of the nominal model is assumed to take values in the initial condition set $\mathscr{X}_{0,\Gamma}$. Note that the dynamics of the nominal and perturbed model depend on the vectors of parameters $\boldsymbol{\pi}\in\mathscr{P}_\Gamma$ and $\tilde{\boldsymbol{\pi}}\in\mathscr{P}_{\tilde{\Gamma}}$, where each of these sets allows for parametric uncertainty. Finally, assume also that for all $\boldsymbol{x}_0\in\mathscr{X}_{0,\Gamma}$ and all $\boldsymbol{\pi},\tilde{\boldsymbol{\pi}}\in\mathscr{P}_\Gamma,\mathscr{P}_{\tilde{\Gamma}}$, each state $\boldsymbol{x}(t),\tilde{\boldsymbol{x}}(t)\to\boldsymbol{0}$ as $t\to\infty$.

A reduced model is a specific example of a perturbed model; in this thesis, such a perturbation will be called a *structural perturbation.* Other possible perturbations include: *parametric perturbations,* where $\tilde{N}=N$ and $\tilde{\boldsymbol{f}}=\boldsymbol{f}$, but one or more of the parameters $\tilde{\boldsymbol{\pi}}\neq\boldsymbol{\pi}$ are perturbed; *parameter uncertainty,* where $\mathscr{P}_\Gamma=\{\boldsymbol{\pi}\}$ is not uncertain, but $\mathscr{P}_{\tilde{\Gamma}}\supset\mathscr{P}_\Gamma$ is; *retroactive effects,* where the form of the vector field $\tilde{\boldsymbol{f}}$ is affected by a downstream system; and so on. For each type of perturbation, the dynamic response of the trajectory from a given initial condition $\boldsymbol{x}_0$ is measured by $\Delta\big(\Gamma(\boldsymbol{x}_0),\tilde{\Gamma}(\tilde{\boldsymbol{x}}_0)\big)$.

Notwithstanding these additional applications, the task of estimating $\Delta\big(\Gamma(\boldsymbol{x}_0),\tilde{\Gamma}(\tilde{\boldsymbol{x}}_0)\big)$ will be called the *error estimation* problem, reflecting model order reduction as the primary motivation of this chapter. Of course, given a particular initial condition and parameter vector, it may be possible for each system in (3.1) to be simulated and the error quantified. However, the dimension N of the model may be large, or the dynamics may exist at many timescales, such that simulation is impractical or inaccurate. Moreover, the initial conditions $\boldsymbol{x}_0$ and parameters $\boldsymbol{\pi},\tilde{\boldsymbol{\pi}}$ may vary across high-dimensional sets, requiring an impractically large number of simulations in order to find the worst-case value of $\Delta\big(\Gamma(\boldsymbol{x}_0),\tilde{\Gamma}(\tilde{\boldsymbol{x}}_0)\big)$. However, the technique described below avoids simulation by producing an *a priori* upper bound on the worst-case error from a single semi-definite program.

3.1.1 The error system

Assume for now that the perturbed system $\tilde{\Gamma}$ is given, and that neither system is uncertain (hence dropping the $\boldsymbol{\pi}$, $\tilde{\boldsymbol{\pi}}$ notation). The key to the comparison of the models in (3.1) is the construction of a corresponding *error system.* The output of the error system is a trajectory that tracks the difference between the outputs of the nominal and perturbed models (i.e. the full and reduced models, in the case of model reduction) at each time point.

Definition 3.1 For the two models given in (3.1), the *error system* is

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x}),\tag{3.2a}$$

$$\dot{\tilde{\boldsymbol{x}}}=\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}),\tag{3.2b}$$

$$\boldsymbol{y}_e=\boldsymbol{h}(\boldsymbol{x})-\tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}}),\tag{3.2c}$$

$$\boldsymbol{x}_e(0)^T=\big(\boldsymbol{x}_0^T,\boldsymbol{\chi}(\boldsymbol{x}_0)^T\big)\in\mathscr{X}_0,\tag{3.2d}$$

with state $\boldsymbol{x}_e^T = [\boldsymbol{x}^T, \tilde{\boldsymbol{x}}^T]$ taking values $\boldsymbol{x}_e(t) \in \mathscr{D}$ for the state space $\mathscr{D} = \mathscr{D}_\Gamma \times \mathscr{D}_{\tilde{\Gamma}} \subseteq \mathbb{R}_{\geq 0}^{N+\tilde{N}}$. Following from the assumptions on (3.1), we assume that $\boldsymbol{x}_e(t) \to \mathbf{0} \in \mathscr{D}$ as $t \to \infty$ for all possible initial conditions $\boldsymbol{x}_e(0) \in \mathscr{X}_0$. ◊

The central theme of this chapter is the use of SOS programming for the error estimation problem. The following result is essentially a special case of Example 2.4 applied to the error system, and therefore links dissipativity theory to the estimation of the error. Combined with the results of Chapter 2, this result is therefore the key to the computational technique for the error estimation problem.

Proposition 3.2. *Suppose that a storage function $V(\boldsymbol{x}_e)$ exists, satisfying* (2.2) *and thereby certifying the dissipativity of the error system* (3.2) *with respect to the supply rate $s(\boldsymbol{y}_e) = -\boldsymbol{y}_e^T\boldsymbol{y}_e$. Fix an arbitrary initial condition $\boldsymbol{x}_e(0) = \big(\boldsymbol{x}_0, \boldsymbol{\chi}(\boldsymbol{x}_0)\big) \in \mathscr{X}_0$. Then the $\mathscr{L}_2$ norm of the error trajectory corresponding to this initial condition $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2$ is finite, and satisfies*

$$\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2^2 \leq V\big(\boldsymbol{x}_e(0)\big). \tag{3.3a}$$

Furthermore, the worst-case error between the two systems is bounded above by the value of the storage function

$$\max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} \Delta\big(\Gamma(\boldsymbol{x}_0), \tilde{\Gamma}(\tilde{\boldsymbol{x}}_0)\big) = \max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2^2 \leq \max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} V\big(\boldsymbol{x}_e(0)\big), \tag{3.3b}$$

maximised over the initial condition set $\boldsymbol{x}_e(0) \in \mathscr{X}_0$. In fact, since $\boldsymbol{x}_e(0)^T = \big[\boldsymbol{x}_0^T, \boldsymbol{\chi}(\boldsymbol{x}_0)^T\big]$, this quantity can be maximised over $\boldsymbol{x}_0 \in \mathscr{X}_{0,\Gamma}$.

Proof. The dissipativity of (3.2) implies that there exists a storage function $V(\boldsymbol{x}_e)$ such that (2.2) and thus equivalently (2.3) holds for any fixed, arbitrary $\boldsymbol{x}_e(0) \in \mathscr{X}_0$. Substituting $s(\boldsymbol{y}_e)$ into (2.3) gives

$$V\big(\boldsymbol{x}_e(T_2)\big) \leq V\big(\boldsymbol{x}_e(T_1)\big) - \int_{T_1}^{T_2} |\boldsymbol{y}(t) - \tilde{\boldsymbol{y}}(t)|^2 \,\mathrm{d}t.$$

As $T_1 \to 0$ and $T_2 \to \infty$ we find that

$$\int_0^\infty |\boldsymbol{y}(t) - \tilde{\boldsymbol{y}}(t)|^2 \,\mathrm{d}t + V(\mathbf{0}) \leq V\big(\boldsymbol{x}_e(0)\big) \leq \max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} V\big(\boldsymbol{x}_e(0)\big), \tag{3.4}$$

where $\boldsymbol{x}_e(T_2) \to \mathbf{0}$ in the limit as $T_2 \to \infty$, by assumption. Note also that (2.2) for $s = -\boldsymbol{y}_e^T\boldsymbol{y}_e$ implies that $\dot{V} \leq 0$, and hence that V is a Lyapunov function for the error system with $\arg\min(V) = \mathbf{0}$. Hence we can assume without loss of generality that $V(\mathbf{0}) = 0$. The result in (3.3a) follows from the first inequality in (3.4), and (3.3b) follows from the second. □

Proposition 3.2 identifies the search for an upper bound on $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2$ with the existence of a storage function certifying the dissipativity of the error system in Definition 3.1 with respect to $s(\boldsymbol{y}_e) = -\boldsymbol{y}_e^T\boldsymbol{y}_e$. In Chapter 2, SOS programming was applied to implement the search for a storage function as an optimisation program, in the case of rational polynomial dynamical systems. As described in Chapter 1, BRN models and their reductions usually fall into this category, and are therefore amenable to SOS programming techniques.

3.1.2 Fixed initial conditions

The following SOS program can be implemented in SOSTOOLS to construct an upper bound on the $\mathscr{L}_2$ error incurred by a given model perturbation. For simplicity, we first assume that the set of initial conditions $\mathscr{X}_0 = \{\boldsymbol{x}_{e,0}\}$ is a single vector. We also assume that the state space $\mathscr{D} \subseteq \mathbb{R}^{N+\tilde{N}}_{\geq 0}$ can be approximated by

$$\mathscr{D} \subseteq \left\{\boldsymbol{x}_e \in \mathbb{R}^{N+\tilde{N}} \mid \phi_i(\boldsymbol{x}_e) \leq 0 \text{ for all } i = 1,\dots,R\right\} = \Phi$$

for a finite collection of given polynomials $\phi_1,\dots,\phi_R \in \mathbb{R}[\boldsymbol{x}_e]$.

Program 3.3 Choose the $R+1$ polynomials $V, \sigma_i \in \mathbb{R}[\boldsymbol{x}_e]$, where $i = 1,\dots,R$, in order to minimise $V\big(\boldsymbol{x}_e(0)\big)$ subject to the constraints

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.5a}$$

$$\sigma_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.5b}$$

$$\begin{aligned}-\psi(\boldsymbol{x}_e)\left[\boldsymbol{y}_e^T\boldsymbol{y}_e + \nabla_{\boldsymbol{x}}V\cdot\boldsymbol{f}(\boldsymbol{x}) + \nabla_{\tilde{\boldsymbol{x}}}V\cdot\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}})\right]&\\ + \sum_{i=1}^{R}\sigma_i(\boldsymbol{x}_e)\phi_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e],&\end{aligned} \tag{3.5c}$$

where we assume that the common denominator $\psi(\boldsymbol{x}_e) > 0$ of the rational polynomial dynamics is positive in $\mathscr{D}$, and $\nabla_{\boldsymbol{x}}V$ denotes the partial Jacobian of $V(\boldsymbol{x}_e) = V(\boldsymbol{x},\tilde{\boldsymbol{x}})$ with respect to $\boldsymbol{x}$ (and similarly for $\tilde{\boldsymbol{x}}$). ◊

This SOS program is a special case of Program 2.13. Hence, by application of Proposition 2.14, it returns a storage function V that certifies the dissipativity of the error system with respect to $s(\boldsymbol{y}_e) = -\boldsymbol{y}_e^T\boldsymbol{y}_e$. Furthermore, Proposition 3.2 implies that $\|\boldsymbol{y}_e\|^2 \leq V(\boldsymbol{x}_{e,0})$. Since $V(\boldsymbol{x}_{e,0})$ is minimised by Program 3.3, the resulting SOS storage function V supplies the tightest upper error bound $\|\boldsymbol{y}_e\|^2 \leq V(\boldsymbol{x}_{e,0})$ possible for the output $\boldsymbol{y}_e$ of the error system (3.2) from the fixed initial condition $\boldsymbol{x}_{e,0}$.

3.1.3 Worst-case error

The formulation of Program 3.3 assumes that the initial conditions are fixed. However, suppose the set $\mathscr{X}_0$ of possible initial conditions is a strict superset of $\{\boldsymbol{x}_{e,0}\}$. The largest value over $\boldsymbol{x}_e(0) \in \mathscr{X}_0$ of the storage function $V\big(\boldsymbol{x}_e(0)\big)$ returned by Program 3.3 will not, in general, provide the tightest upper bound on the *worst-case* error. The objective in this case should instead be to minimise $\max_{\boldsymbol{x}_e\in\mathscr{X}_0} V(\boldsymbol{x}_e)$. As discussed in Section 2.3.4, one possible heuristic approach to this task is to adapt Program 3.3 to instead minimise trace$(\boldsymbol{Q}_V)$, where $\boldsymbol{Q}_V$ is the decision variable matrix used to define V in the SOS expression $V(\boldsymbol{x}_e) = \boldsymbol{z}(\boldsymbol{x}_e)^T\boldsymbol{Q}_V\boldsymbol{z}(\boldsymbol{x}_e)$. Denote the resulting storage function V_{tr}. Once V_{tr} has been returned, it is a further challenge to actually estimate the largest value of V_{tr} in $\mathscr{X}_0$. We now consider an alternative approach to finding a storage function V that simultaneously estimates and minimises the upper bound $\max_{\boldsymbol{x}_e\in\mathscr{X}_0} V(\boldsymbol{x}_e)$ on the worst-case error.

Recall the construction of the estimate $\mathscr{D} \subseteq \Phi$ above. The following SOS program is an adaptation of Program 3.3 that makes use of a similar estimate for the set of initial conditions

$$\mathscr{X}_0 \subseteq \left\{ \boldsymbol{x}_e \in \mathbb{R}^{N+\tilde{N}} \;\middle|\; \begin{array}{c} \theta_i(\boldsymbol{x}_e) \le 0 \;\forall i = 1, \dots, R_1 \\ \theta_i(\boldsymbol{x}_e) = 0 \;\forall i = R_1+1, \dots, R_1+R_2 \end{array} \right\} = \Theta,$$

for the $R_1 + R_2$ polynomials $\theta_i \in \mathbb{R}[\boldsymbol{x}_e]$.

Program 3.4 Choose the $R_1 + R_2 + R + 1$ polynomials $V, \sigma_i, \tau_j \in \mathbb{R}[\boldsymbol{x}_e]$, where $i = 1, \dots, R$ and $j = 1, \dots, R_1 + R_2$, in order to minimise the squared constant decision variable B^2 subject to the constraints (3.5), and the additional constraints

$$\tau_j(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e] \text{ for } j = 1, \dots, R_1, \tag{3.6a}$$

$$\left[B^2 - V(\boldsymbol{x}_e)\right] + \sum_{j=1}^{R_1+R_2} \tau_j(\boldsymbol{x}_e)\theta_j(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e]. \tag{3.6b}$$

Note that $\tau_j(\boldsymbol{x}_e)$ is not necessarily an SOS polynomial for $j > R_1$. ◊

Proposition 3.5. *The value of B^2 returned by Program 3.4 is an upper bound on the worst-case error* $\max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} \|\boldsymbol{y}_e\|^2$.

Proof. Since V satisfies (3.5), it follows that for each fixed initial condition $\boldsymbol{x}_e(0) \in \mathscr{X}_0$ the bound $\|\boldsymbol{y}_e\|^2 \le V(\boldsymbol{x}_e(0))$ holds. The additional constraints (3.6) imply that, for each initial condition $\boldsymbol{x}_e(0) \in \mathscr{X}_0$,

$$B^2 - V(\boldsymbol{x}_e(0)) \ge \left[B^2 - V(\boldsymbol{x}_e(0))\right] + \sum_{j=1}^{R_1+R_2} \tau_j(\boldsymbol{x}_e(0))\theta_j(\boldsymbol{x}_e(0)) \ge 0,$$

and hence B^2 is an upper bound on the value of $V(\boldsymbol{x}_e(0)) \ge \|\boldsymbol{y}_e\|^2$. Since B^2 is constant over varying $\boldsymbol{x}_e(0) \in \mathscr{X}_0$, it follows that $B^2 \ge \max_{\boldsymbol{x}_e(0)\in\mathscr{X}_0} \|\boldsymbol{y}_e\|^2$. □

Clearly, by minimising B^2, Program 3.4 is a relaxation which, although suboptimal, may provide a relatively tight bound on the worst-case error. If the storage function is the output of an SOS program, it must be maximised over $\mathscr{X}_0$ in order to find an upper bound on the worst-case error. However, the upper bound B^2 is directly calculated as an output of Program 3.4, and hence the additional task of maximising the storage function V is avoided. We will demonstrate the use of this approach to bounding the worst-case error in the following example.

3.2 Example: Model Reduction of Enzyme Kinetics

This section demonstrates the use of the SOS error-bounding technique described above in a classical example of the model order reduction of a BRN [133]. The enzyme kinetics network consists of $M = 3$ reactions between a total of $N = 4$ reactants: substrate X, product Y, enzyme E, and enzyme–substrate complex C. This network is written in the form of (1.1) as

$$X + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C \xrightarrow{k_2} Y + E. \tag{3.7}$$

Let lower-case letters correspond to concentrations, so that $\big(x(t), c(t), e(t), y(t)\big)$ are the concentrations of the reactants (X, C, E, Y) at time t. The dynamics of this network can be modelled in the form of (1.2) as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \\ \dot{e}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 x(t) e(t) \\ k_{-1} c(t) \\ k_2 c(t) \end{bmatrix},$$

with initial conditions $\big(x(0), c(0), e(0), y(0)\big) = (x_0, c_0, e_0, 0)$. Note that we can assume $y(0) = 0$ without loss of generality, since the dynamics are independent of $y(t)$.

Since this stoichiometry has rank($\boldsymbol{S}$) = 2, this system can be expressed as an ODE with state in $\mathbb{R}^2_{\geq 0}$ without incurring any approximation error (see Section 4.3.1 for more details). There are two independent conservation relations: by summing the corresponding rows in $\boldsymbol{S}$, we find that $\dot{c} + \dot{e} = 0$ and $\dot{x} + \dot{c} + \dot{y} = 0$. The two conserved constants $E_0 = c_0 + e_0$ and $X_0 = x_0 + c_0$ are parameters that denote the total (i.e. free and bound) enzyme concentration and the total (initial) substrate concentration, respectively. Then we can write the enzyme kinetics example as a two-dimensional system with dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x(t)\,[E_0 - c(t)] \\ k_{-1} c(t) \\ k_2 c(t) \end{bmatrix}, \tag{3.8}$$

and initial conditions $\big(x(0), c(0)\big) = (x_0, c_0)$, where the other two species have concentrations

$$\begin{aligned} e(t) &= E_0 - c(t), \\ y(t) &= X_0 - x(t) - c(t). \end{aligned}$$

The state space $\mathscr{D}_\Gamma$ of this system is constrained such that $0 \leq c(t) \leq E_0$, and $0 \leq x(t) \leq X_0 - c(t)$. For fixed X_0, the set $\mathscr{X}_{0,\Gamma}$ of initial conditions $(x_0, c_0) \in \mathscr{X}_{0,\Gamma} \subseteq \mathscr{D}_\Gamma$ is further constrained such that $x_0 + c_0 = X_0$, hence parametrising the initial conditions as $(x_0, X_0 - x_0)$ for $x_0 \in \big[(X_0 - E_0)^+, X_0\big]$. For all initial conditions, the state $(x, c) \to (0, 0)$ as time $t \to \infty$. Thus, the trajectory of this system tracks the conversion of X_0 units of (free and bound) substrate into X_0 units of product, through the transient binding of E_0 units of enzyme.

3.2.1 Quasi-steady state approximations

The enzyme dynamics network (3.7) with dynamics (3.8) is traditionally approximated using *singular perturbation*, employing various types of quasi-steady state approximation [116, 133, 205]. Theorem 1.1 (i.e. Tikhonov's Theorem [108, 121]) is the underlying paradigm for singular perturbation. If a parameter $\mu \ll 1$ can be extracted from the model such that the enzyme kinetics (3.8) can be written in the form of (1.5), then a reduction of the form (1.6) will be a good approximation, in the sense of having little error. Much of the debate about the best reduction of the enzyme kinetics model [175] can be characterised as ensuring that the dynamics $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ on the right-hand side of (1.5) are fully independent of the parameter $\mu \ll 1$.

The classical approach to this problem assumes that the total enzyme concentration E_0 is on a much smaller scale than the total substrate concentration X_0. The parameter $\mu = E_0/X_0 \ll 1$ can then be used in a non-dimensionalisation of the system, that results in a re-writing of the dynamics in the scale-separated form (1.5). Reducing the non-dimensional system to (1.6) according to Tikhonov's Theorem, and re-writing in the original dimensional terms again, gives the one-dimensional system

$$\dot{\tilde{x}}(t) = -\frac{k_2 E_0 \tilde{x}(t)}{\tilde{x}(t) + K_m} \tag{3.10a}$$

for constant $K_m = (k_{-1} + k_2)/k_1$, with the other species given by the algebraic relationship

$$\tilde{c}(t) = \frac{E_0 \tilde{x}(t)}{\tilde{x}(t) + K_m}, \tag{3.10b}$$

and both $\tilde{e}(t)$ and $\tilde{y}(t)$ found by replacing x and c with $\tilde{x}$ and $\tilde{c}$ in (3.9) above. The reduction (3.10) is known as the standard quasi-steady state approximation, or sQSSA. For more details on this reduction, see [133].

The assumption of $E_0 \ll X_0$ for the sQSSA implies that c is a much faster variable than x and therefore quickly equilibrates, resulting in the expression (3.10b). An alternative approach is to consider the case where x is the fast variable, and c is slow. This is known as the reverse QSSA (rQSSA, [180]), and results in the reduced one-dimensional model given by

$$\dot{\tilde{c}}(t) = -k_2 \tilde{c}(t) \tag{3.11a}$$

with the other species given by

$$\tilde{x}(t) = K\left(\frac{\tilde{c}(t)}{E_0 - \tilde{c}(t)}\right), \text{ for } K = \frac{k_{-1}}{k_1}, \tag{3.11b}$$

where for $\tilde{x}(t)$ to be well-defined we have to ensure that $\tilde{c}(t) < E_0$. The initial condition of the reduced model is set to $\tilde{c}(0) = E_0 X_0/(E_0 + X_0 + K)$, independently of (x_0, c_0). Note that, again, both $\tilde{e}(t)$ and $\tilde{y}(t)$ can be found by substituting x and c with $\tilde{x}$ and $\tilde{c}$ in (3.9) above.

A third approximation approach is known as the total QSSA, or tQSSA. This is justified by the observation that it is often reactions which can be classified as fast or slow, rather than species [116, 151, 152]. In the case of the tQSSA, it is assumed that the total concentration $\xi = x + c$ is the slow variable, and that c is fast. The resulting approximation [28, 48, 205] is given by the dynamics

$$\dot{\tilde{\xi}}(t) = -k_2 \tilde{c}(t) \tag{3.12a}$$

with the initial condition $\xi(0) = X_0$ and $\tilde{c}$ approximated by

$$\tilde{c}(t) = \frac{E_0 \tilde{\xi}(t)}{E_0 + K_m + \tilde{\xi}(t)}. \tag{3.12b}$$

Each initial condition $(x_0, X_0 - x_0) \in \mathscr{X}_{0,\Gamma}$ for the original model always results in the same initial condition $\mathscr{X}_{0,\tilde{\Gamma}} = \{X_0\}$ for the tQSSA dynamics. The substrate concentration $\tilde{x}(t) = \tilde{\xi}(t) - \tilde{c}(t)$ can be deduced, and again we can apply (3.9) to find $\tilde{e}(t)$ and $\tilde{y}(t)$. Further to these reductions, Chapter 5 will provide an expression for the approximated dynamics in the case where the variable $c(t)$ is neither fast nor slow.

Consider the four parameter vector values $\boldsymbol{\pi}_i$, $i = 1,2,3,4$, given in Table 3.1. Figure 3.1 depicts, for each $\boldsymbol{\pi}_i$ and for the fixed initial condition with $x_0 = X_0$, the trajectory of x and the

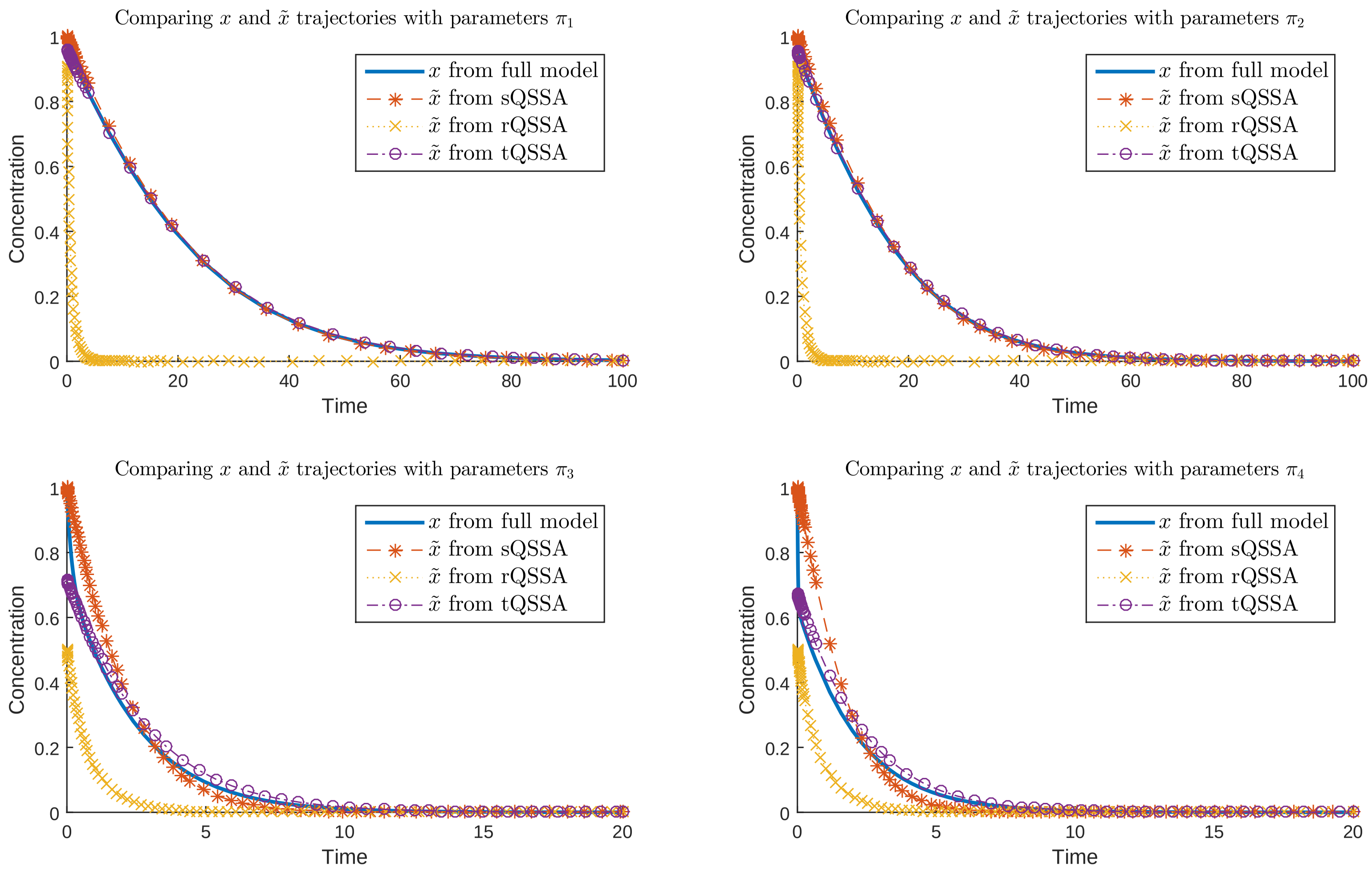


Figure 3.1: Simulations of enzyme kinetics. The solid curve tracks the trajectory of x simulated by the full model (3.8). The broken curves track the trajectories of $\tilde{x}$ simulated by the QSSA-reduced models (3.10), (3.11), and (3.12). Parameters $\boldsymbol{\pi}_i$ and estimated error norms $\|x - \tilde{x}\|^2$ are given in Table 3.1. Simulations were carried out using the MATLAB ODE solver `ode15s`.

	$\boldsymbol{\pi}_1$	$\boldsymbol{\pi}_2$	$\boldsymbol{\pi}_3$	$\boldsymbol{\pi}_4$
X_0	1	1	1	1
E_0	0.1	0.1	1	1
k_1	2	20	2	20
k_{-1}	2	20	2	20
k_2	1	1	1	1
	Simulated $\|x-\tilde{x}\|^2$			
sQSSA	0.0078	0.0153	0.0465	0.0906
rQSSA	9.3864	7.0340	0.3527	0.1639
tQSSA	0.0013	0.0017	0.0118	0.0082

Table 3.1: Fixed parameter values and squares of simulated error on applying QSSA model reduction. The values of $\|x-\tilde{x}\|^2$ were estimated from the simulated trajectories in Figure 3.1 using MATLAB function `trapz`, evaluated on the vector of time and states output from `ode15s`.

three trajectories $\tilde{x}$ of each QSSA reduction. Clearly the validity of each reduction, in terms of the simulated errors $\|x-\tilde{x}\|^2$, varies with the parameter values. There has been a great deal of work to determine the various regions of parameter space in which each of the three varieties of QSSA listed above are valid [48, 175, 205]. In the rest of this section, we will apply the computational techniques for error bounding to compare each reduction (3.10), (3.11), and (3.12) with the full-scale model (3.8).

3.2.2 Error quantification

This section describes the use of SOS programming to compare the trajectories x and $\tilde{x}$ in the nominal model and each of the three reduced models. Table 3.1 records simulations of the error for fixed initial conditions and parameter values. As described in Section 3.1, the comparison can instead be carried out by SOS programming. We first consider the example in this section with fixed initial conditions and parameters, as a proof of principle. Section 3.2.3 will proceed to demonstrate the value of the SOS approach by finding a worst-case error across all initial conditions without the need for exhaustive simulation.

The first step is to identify each of the error systems corresponding to the full and QSSA-reduced model by substituting (3.8) and each of (3.10), (3.11), and (3.12) into (3.2). Second, the dynamics of each error system are substituted into Program 3.3 to return V satisfying Proposition 3.2.

3.2.2.1 sQSSA

Consider the state $\boldsymbol{x}_e = (x, c, \tilde{x})^T$ with dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x(t)\,[E_0 - c(t)] \\ k_{-1} c(t) \\ k_2 c(t) \end{bmatrix}, \tag{3.13a}$$

$$\dot{\tilde{x}}(t) = -\frac{k_2 E_0 \tilde{x}(t)}{\tilde{x}(t) + K_m}, \tag{3.13b}$$

$$\boldsymbol{y}_e(t) = x(t) - \tilde{x}(t) \tag{3.13c}$$

$$\boldsymbol{x}_e(0) = (X_0, 0, X_0)^T, \tag{3.13d}$$

where the state takes values $\boldsymbol{x}_e(t) \in \mathscr{D} \subseteq \Phi$ bounded by the estimate $\Phi = \{\boldsymbol{x}_e \mid \phi_i(\boldsymbol{x}_e) \leq 0\}$ for the polynomials

$$\phi_1(\boldsymbol{x}_e) = x(x + c - X_0), \tag{3.13e}$$

$$\phi_2(\boldsymbol{x}_e) = c(c - E_0), \tag{3.13f}$$

$$\phi_3(\boldsymbol{x}_e) = \tilde{x}(\tilde{x} - X_0). \tag{3.13g}$$

This error system is used to construct the following SOS program.

Program 3.6 Minimise $V(X_0, 0, X_0)$ over the four polynomials $V, \sigma_i \in \mathbb{R}[x, c, \tilde{x}]$, for $i = 1, \ldots, 3$, subject to the SOS conditions

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.14a}$$

$$\sigma_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.14b}$$

$$\begin{aligned} &-(\tilde{x} + K_m)\Big[(x - \tilde{x})^2 + \frac{\partial V}{\partial x}\big(k_{-1}c - k_1 x(E_0 - x)\big) \\ &+ \frac{\partial V}{\partial c}\big(k_1 x(E_0 - x) - (k_{-1} + k_2)c\big) - \frac{\partial V}{\partial \tilde{x}}\left(\frac{k_2 E_0 \tilde{x}}{\tilde{x} + K_m}\right)\Big] \\ &+ \sum_{i=1}^{3} \sigma_i(\boldsymbol{x}_e)\phi_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \end{aligned} \tag{3.14c}$$

where ϕ_i for $i = 1, 2, 3$ are given in (3.13). ◊

3.2.2.2 rQSSA

Consider the state $\boldsymbol{x}_e = (x, c, \tilde{c})^T$ with dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x(t)\,[E_0 - c(t)] \\ k_{-1} c(t) \\ k_2 c(t) \end{bmatrix}, \tag{3.15a}$$

$$\dot{\tilde{c}}(t) = -k_2 \tilde{c}(t), \tag{3.15b}$$

$$\boldsymbol{y}_e(t) = x(t) - \frac{k_{-1}}{k_1}\left(\frac{\tilde{c}(t)}{E_0 - \tilde{c}(t)}\right) \tag{3.15c}$$

$$\boldsymbol{x}_e(0) = (X_0, 0, C_0)^T, \tag{3.15d}$$

for the parameter-dependent value

$$C_0 = \frac{E_0 X_0}{E_0 + X_0 + (k_{-1}/k_1)}$$

of the initial condition $\tilde{c}(0)$. Here the state takes values $\boldsymbol{x}_e(t) \in \mathscr{D} \subseteq \Phi$ bounded by the polynomial estimate $\Phi = \{\boldsymbol{x}_e \mid \phi_i(\boldsymbol{x}_e) \leq 0\}$ for the polynomials

$$\phi_1(\boldsymbol{x}_e) = x(x + c - X_0), \tag{3.15e}$$

$$\phi_2(\boldsymbol{x}_e) = c(c - E_0), \tag{3.15f}$$

$$\phi_3(\boldsymbol{x}_e) = \tilde{c}(\tilde{c} - C_0). \tag{3.15g}$$

This error system is used to construct the following SOS program.

Program 3.7 Minimise $V(X_0, 0, C_0)$ over the four polynomials $V, \sigma_i \in \mathbb{R}[x, c, \tilde{c}]$, for $i = 1, \ldots, 3$, subject to the SOS conditions

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.16a}$$

$$\sigma_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.16b}$$

$$\begin{aligned} -(E_0 - \tilde{c})^2 \Bigg[\left(x - \frac{k_{-1}\tilde{c}}{k_1(E_0 - \tilde{c})} \right)^2 &+ \frac{\partial V}{\partial x}\big(k_{-1}c - k_1 x(E_0 - x)\big) \\ + \frac{\partial V}{\partial c}\big(k_1 x(E_0 - x) - (k_{-1} + k_2)c\big) &- \frac{\partial V}{\partial \tilde{c}} k_2 \tilde{c} \Bigg] \\ &+ \sum_{i=1}^{3} \sigma_i(\boldsymbol{x}_e)\phi_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \end{aligned} \tag{3.16c}$$

where ϕ_i for $i = 1, 2, 3$ are given in (3.15). ◊

3.2.2.3 tQSSA

Consider the state $\boldsymbol{x}_e = (x, c, \tilde{\xi})^T$ with dynamics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x(t)\,[E_0 - c(t)] \\ k_{-1}c(t) \\ k_2 c(t) \end{bmatrix}, \tag{3.17a}$$

$$\dot{\tilde{\xi}}(t) = -\frac{k_2 E_0 \tilde{\xi}(t)}{E_0 + K_m + \tilde{\xi}(t)}, \tag{3.17b}$$

$$\boldsymbol{y}_e(t) = x(t) - \left(\xi(t) - \frac{E_0 \tilde{\xi}(t)}{E_0 + K_m + \tilde{\xi}(t)} \right) \tag{3.17c}$$

$$\boldsymbol{x}_e(0) = (X_0, 0, X_0)^T, \tag{3.17d}$$

where the state takes values $\boldsymbol{x}_e(t) \in \mathscr{D} \subseteq \Phi$ bounded by the polynomial estimate $\Phi = \{\boldsymbol{x}_e \mid \phi_i(\boldsymbol{x}_e) \leq 0\}$ for the polynomials

$$\phi_1(\boldsymbol{x}_e) = x(x + c - X_0), \tag{3.17e}$$

$$\phi_2(\boldsymbol{x}_e) = c(c - E_0), \tag{3.17f}$$

$$\phi_3(\boldsymbol{x}_e) = \tilde{\xi}(\tilde{\xi} - X_0). \tag{3.17g}$$

This error system can be used to construct the following SOS program.

Program 3.8 Minimise $V(X_0, 0, X_0)$ over the four polynomials $V, \sigma_i \in \mathbb{R}[x, c, \tilde{\xi}]$, for $i = 1, \ldots, 3$, subject to the SOS conditions

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.18a}$$

$$\sigma_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.18b}$$

$$\begin{aligned} -\left(E_0 + K_m + \tilde{\xi}\right)^2 \Bigg[\left(x - \tilde{\xi} + \frac{E_0 \tilde{\xi}}{E_0 + K_m + \tilde{\xi}} \right)^2 + \frac{\partial V}{\partial x} \big(k_{-1} c - k_1 x (E_0 - x) \big) \\ + \frac{\partial V}{\partial c} \big(k_1 x (E_0 - x) - (k_{-1} + k_2) c \big) - \frac{\partial V}{\partial \tilde{\xi}} \left(\frac{k_2 E_0 \tilde{\xi}}{E_0 + K_m + \tilde{\xi}} \right) \Bigg] \\ + \sum_{i=1}^{3} \sigma_i(\boldsymbol{x}_e) \phi_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \end{aligned} \tag{3.18c}$$

where ϕ_i for $i = 1, 2, 3$ are given in (3.17). ◊

3.2.2.4 Results

Each of the SOS programs above has been implemented in SOSTOOLS. The code for a function that returns the upper bound on the sQSSA error, given a fixed parameter vector and a fixed $\deg V$, can be found in Appendix A. The results of applying the SOS method to the error bound estimation can be seen in Table 3.2. These results are comparable to the *a posteriori* bounds found through simulation recorded in Table 3.1.

It can be seen from these results that increasing the degree of the storage function being sought has the effect of increasing the dimension of the SOS program, and therefore increases the required computation time. However, the resulting bound from a higher-degree V is often more precise than a lower-degree V. In fact, the estimates of the error $\|x - \tilde{x}\|^2$ given in Table 3.1 are not necessarily exact values, or even lower bounds, due to numerical imprecision of the MATLAB function `trapz`. Hence, the SOS bounds given in Table 3.2 can in some cases give tighter upper bounds on the error than those in Table 3.1.

Importantly, by being able to quantify the error incurred by a reduction, we have a more precise definition of that reduction's validity. It is well-known [133] from Tikhonov's theorem that, for example, the sQSSA becomes more valid as E_0/X_0 approaches zero. However, comparing the results in the top part of Table 3.2 for $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ demonstrate that the same ratio of E_0/X_0 can produce twice the value of $\|x - \tilde{x}\|^2$, depending on the other parameter values[1]. Therefore a

[1] The sQSS reduction of [180] uses the alternative small parameter $E_0/(K_m + X_0)$ in a different non-dimensionalisation and reduction. The value of K_m decreases from 1.5 at $\boldsymbol{\pi}_1$ to 1.05 at $\boldsymbol{\pi}_2$, suggesting that the sQSSA becomes less valid under this different non-dimensionalisation.

more practical definition of a reduction's validity should be based on whether the incurred error is guaranteed to be below a pre-determined threshold value. Note also that we may prefer to judge the validity of the reduction based on the errors of other output trajectories, such as $\|c-\tilde{c}\|$ or $\|y-\tilde{y}\|$, which is likely to change any conclusions about each reduction's validity.

3.2.3 Worst-case error

The preceding section considers the case where the nominal model has fixed initial conditions $(x_0, c_0) = (X_0, 0)$. However, when defining this system we assumed that the initial conditions could take values $(x_0, X_0 - x_0)$ for values of $x_0 \in \left[(X_0 - E_0)^+, X_0\right]$. Each model reduction technique maps this initial condition to its own initial condition in a specific way: sQSSA has the initial condition $\tilde{x}(0) = x_0$; rQSSA has the initial condition $\tilde{c}(0) = \frac{E_0 x_0}{E_0 + x_0 + (k_{-1}/k_1)}$; and tQSSA has the initial condition $\tilde{\xi}(0) = X_0$.

As discussed in Section 3.1.3, by allowing the initial conditions $\boldsymbol{x}_e(0) \in \mathscr{X}_0$ to vary, a heuristic approach to optimising the upper bound

$$\max_{\boldsymbol{x}_e(0)} \|x-\tilde{x}\|^2 \leq \max_{\boldsymbol{x}_e(0)} V\big(\boldsymbol{x}_e(0)\big)$$

is given by Program 3.4. We will consider, for concreteness, the sQSSA case with parameter values given by $\boldsymbol{\pi}_1$. The estimate of $\mathscr{X}_0 \subseteq \Theta = \{\boldsymbol{x}_e \mid \theta_1 \leq 0, \theta_2 = \theta_3 = 0\}$ is defined by the polynomials

$$\begin{aligned}
\theta_1(\boldsymbol{x}_e) &= \big(x - (X_0 - E_0)^+\big)\big(x - X_0\big),\\
\theta_2(\boldsymbol{x}_e) &= x + c - X_0,\\
\theta_3(\boldsymbol{x}_e) &= \tilde{x} - x.
\end{aligned}$$

These polynomials can be used to construct the following example of Program 3.4.

Program 3.9 Find the constant B^2 and 7 polynomials $V, \sigma_i, \tau_j \in \mathbb{R}[\boldsymbol{x}_e]$ for $i, j = 1,2,3$, such that B^2 is minimised subject to the constraints (3.14), together with

$$\tau_1(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.19a}$$

$$B^2 - V(\boldsymbol{x}_e) + \sum_{j=1}^{3} \tau_j(\boldsymbol{x}_e)\theta_j(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e]. \tag{3.19b}$$

◊

The minimal value of B^2 computed by Program 3.9 with parameters $\boldsymbol{\pi}_1$ is 0.0545, and we can conclude that $\|x-\tilde{x}\|^2 \leq 0.0545$ for all values of $x_0 \in [0.9, 1]$. Denote the resulting storage function by $V^\star$. The values $V^\star\big(\boldsymbol{x}_e(0)\big)$ are depicted in Figure 3.2 for initial conditions $\boldsymbol{x}_e(0) = (x_0, X_0 - x_0, x_0)$ for varying $x_0 \in [0.9, 1]$. Also in this figure is a plot of the degree 4 storage function V found by Program 3.6, which was optimised for $x_0 = X_0$. The bound $B^2 = 0.0545$ on the worst-case error found by Program 3.9 is clearly a tighter bound than $\max_{x_0 \in [0.9,1]} V(x_0, X_0 - x_0, x_0) \approx 0.422$. However, for fixed values of $x_0 \approx X_0$, the upper bound given by $V(x_0, X_0 - x_0, x_0)$ is tighter than that given by $V^\star(x_0, X_0 - x_0, x_0)$. The other plot depicted in Figure 3.2 is the storage function V_{tr} found by maximising trace$(\boldsymbol{Q}_V)$ subject to (3.14), which is a widely-used heuristic for optimising stor-

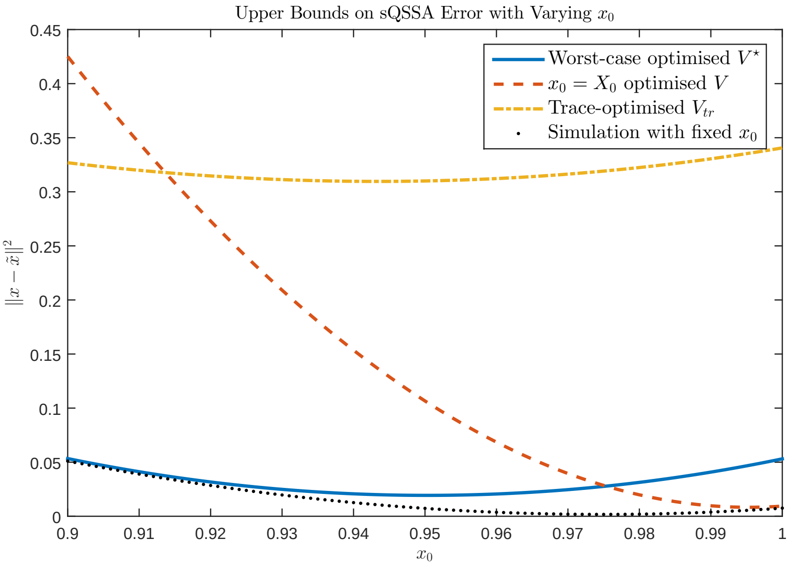


Figure 3.2: Plotting the upper bound values $V\big(\boldsymbol{x}_e(0)\big)$ as $x_0 \in [X_0 - E_0, X_0]$ varies, for three different valid storage functions, constructed respectively through Program 3.6, Program 3.9, and an adaptation of Program 3.6 to minimise trace($\boldsymbol{Q}_V$). Each black point corresponds to an individual simulation of the error for 100 chosen values of $x_0 \in [0.9, 1]$.

age functions. Although V_{tr} gives an improved worst-case error bound over V, it is significantly conservative compared to $V^\star$ as constructed by Program 3.9.

3.3 Example: Parameter Sensitivity

The previous section demonstrated how to use SOS programming to bound the error incurred by the reduction of an enzyme kinetics model. However, the theory of Section 3.1 is applicable to more types of model perturbation than model order reduction. In this section, we consider the response of an output trajectory to a *parametric* perturbation.

Consider the toy BRN consisting of $M = 3$ reactions with $N = 5$ reactants: protein precursors P_1 and P_2; protein polypeptides E_1 and E_2; and the final product, the protein dimer E. For $i = 1, 2$, the reactions are

$$\begin{aligned} P_i &\xrightarrow{\alpha_i} E_i, \\ E_1 + E_2 &\xrightarrow{\alpha_{12}} E, \end{aligned}$$

where the parameters α_1, α_2 and α_{12} define the reaction rates. The model's dynamics can be written in the form of (1.2) as

$$\begin{bmatrix} \dot{p}_1(t) \\ \dot{p}_2(t) \\ \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 p_1(t) \\ \alpha_2 p_2(t) \\ \alpha_{12} e_1(t) e_2(t) \end{bmatrix}, \tag{3.20}$$

with initial conditions $\big(p_1(0), p_2(0), e_1(0), e_2(0), e(0)\big) = (A_1, A_2, 0, 0, 0)$ for the fixed parameters A_1 and A_2. As in the previous example, the conservation relations of this system can be used to rewrite (3.20) in three dimensions as

$$\begin{bmatrix} \dot{p}_1(t) \\ \dot{p}_2(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} -\alpha_1 p_1(t) \\ -\alpha_2 p_2(t) \\ \alpha_{12}\big(A_1 - A_{12} - p_1(t) - \eta(t)\big)\big(A_2 - A_{12} - p_2(t) - \eta(t)\big) \end{bmatrix}, \tag{3.21}$$

where $A_{12} = \min(A_1, A_2)$. This system has initial conditions $\big(p_1(0), p_2(0), \eta(0)\big) = (A_1, A_2, -A_{12})$ and zero steady state. The originally-modelled concentrations can be retrieved by the relations $e(t) = A_{12} + \eta(t)$, and $e_i(t) = A_i - A_{12} - p_i(t) - \eta(t)$.

3.3.1 Parametric perturbation

Assume that all of the parameters of the nominal and perturbed model are known. In terms of (3.1), $\mathscr{P}_\Gamma$ and $\mathscr{P}_{\tilde{\Gamma}}$ are each singleton sets of parameter vectors $\boldsymbol{\pi} \neq \tilde{\boldsymbol{\pi}}$. We can now bound the response of this model to a parametric perturbation by constructing the error system given by Definition 3.1, with perturbed dynamics of the form (3.21) but replacing each α_i with $\tilde{\alpha}_i$. The SOS program corresponding to these parameters is as follows.

Program 3.10 Minimise $V(A_1, A_2, -A_{12}, A_1, A_2, -A_{12})$ over the set of 7 polynomials $V, \sigma_i \in \mathbb{R}[\boldsymbol{x}_e]$ for $i = 1, \ldots, 6$, mapping the state $\boldsymbol{x}_e = (p_1, p_2, \eta, \tilde{p}_1, \tilde{p}_2, \tilde{\eta})$ to $\mathbb{R}$, that satisfy the SOS conditions

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.22a}$$

$$\sigma_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.22b}$$

$$\begin{aligned} -\Big[(\eta - \tilde{\eta})^2 - \alpha_1 \frac{\partial V}{\partial p_1} p_1 - \alpha_2 \frac{\partial V}{\partial p_2} p_2 - \tilde{\alpha}_1 \frac{\partial V}{\partial \tilde{p}_1} \tilde{p}_1 - \tilde{\alpha}_2 \frac{\partial V}{\partial \tilde{p}_2} \tilde{p}_2 \\ + \alpha_{12} \frac{\partial V}{\partial \eta} (p_1 + \eta + A_{12} - A_1)(p_2 + \eta + A_{12} - A_2) \\ + \tilde{\alpha}_{12} \frac{\partial V}{\partial \tilde{\eta}} (\tilde{p}_1 + \tilde{\eta} + A_{12} - A_1)(\tilde{p}_2 + \tilde{\eta} + A_{12} - A_2)\Big] \\ + \sum_{i=1}^{6} \sigma_i(\boldsymbol{x}_e)\phi_i(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \end{aligned} \tag{3.22c}$$

for the polynomials ϕ_i defining the estimate $\mathscr{D} \subseteq \Phi = \{\boldsymbol{x}_e \mid \phi_i(\boldsymbol{x}_e) \leq 0 \ \forall i\}$ given by

$$\begin{aligned} \phi_1(\boldsymbol{x}_e) &= p_1(p_1 + \eta + A_{12} - A_1), & \phi_4(\boldsymbol{x}_e) &= \tilde{p}_1(\tilde{p}_1 + \tilde{\eta} + A_{12} - A_1), \\ \phi_2(\boldsymbol{x}_e) &= p_2(p_2 + \eta + A_{12} - A_2), & \phi_5(\boldsymbol{x}_e) &= \tilde{p}_2(\tilde{p}_2 + \tilde{\eta} + A_{12} - A_2), \\ \phi_3(\boldsymbol{x}_e) &= \eta(\eta + A_{12}), & \phi_6(\boldsymbol{x}_e) &= \tilde{\eta}(\tilde{\eta} + A_{12}). \end{aligned}$$

The resulting $V(A_1, A_2, -A_{12}, A_1, A_2, -A_{12})$ is an upper bound on $\|e - \tilde{e}\|^2$. ◊

For concreteness, take $A_1 = 1$ and $A_2 = 2$, so that $A_{12} = 1$; also set the nominal parameters $\alpha_1 = \alpha_2 = \alpha_{12} = 1$, and the perturbed parameters $\tilde{\alpha}_1 = 2$, leaving $\tilde{\alpha}_2 = \alpha_2$ and $\tilde{\alpha}_{12} = \alpha_{12}$ unperturbed. We have implemented this program in SOSTOOLS. The resulting bound on the output response (i.e. error) is $\|e - \tilde{e}\|^2 \leq 0.02994$, searching over $\deg(V) = 4$ and $\deg(\sigma_i) = 4$. This compares to the error estimate $\|e - \tilde{e}\|^2 \approx 0.0300$ found by simulation of the system. The semi-definite program associated with Program 3.10 has dimension 11,992 and takes 47.96 seconds to complete. When all parameters are fixed, simulation is clearly a faster approach for finding error in this case. The following section considers the case where the reaction parameter $\alpha_{12} = \tilde{\alpha}_{12}$ is uncertain.

3.3.2 Perturbation and uncertainty

Now consider the model (3.21) where the equal parameters $\alpha_{12} = \tilde{\alpha}_{12}$ take an uncertain value $\alpha_{12} \in [\underline{\alpha}_{12}, \bar{\alpha}_{12}]$. Assume that all of the other parameter values remain the same as above, including the perturbation of $\alpha_1 = 1$ to $\tilde{\alpha}_1 = 2$. The extension of Program 3.10 to account for uncertainty follows the form given in Section 2.3.3, where the uncertain parameter is an auxiliary variable. The resulting SOS program is as follows, for the fixed parameter values $\alpha_1 = \alpha_2 = \tilde{\alpha}_2 = 1$, $\tilde{\alpha}_1 = 2$, and $(A_1, A_2) = (1,2)$.

Program 3.11 Minimise $V(1,2,-1,1,2,-1)$ over the set of 8 polynomials $V \in \mathbb{R}[\boldsymbol{x}_e]$ and $\sigma_i \in \mathbb{R}[\boldsymbol{x}_e, \alpha_{12}]$ for $i = 1,\ldots,7$, satisfying the SOS conditions

$$V(\boldsymbol{x}_e) \in \Sigma[\boldsymbol{x}_e], \tag{3.23a}$$

$$\sigma_i(\boldsymbol{x}_e, \alpha_{12}) \in \Sigma[\boldsymbol{x}_e, \alpha_{12}], \tag{3.23b}$$

$$\begin{aligned} -\Big[(\eta-\tilde{\eta})^2 - \frac{\partial V}{\partial p_1}p_1 - \frac{\partial V}{\partial p_2}p_2 - 2\frac{\partial V}{\partial \tilde{p}_1}\tilde{p}_1 - \frac{\partial V}{\partial \tilde{p}_2}\tilde{p}_2 \\ +\alpha_{12}\frac{\partial V}{\partial \eta}(p_1+\eta)(p_2+\eta-1) \\ +\alpha_{12}\frac{\partial V}{\partial \tilde{\eta}}(\tilde{p}_1+\tilde{\eta})(\tilde{p}_2+\tilde{\eta}-1)\Big] \\ +\sum_{i=1}^{7}\sigma_i(\boldsymbol{x}_e, \alpha_{12})\phi_i(\boldsymbol{x}_e, \alpha_{12}) \in \Sigma[\boldsymbol{x}_e, \alpha_{12}], \end{aligned} \tag{3.23c}$$

where the polynomials ϕ_i for $i = 1,\ldots,6$, as defined in Program 3.10 above, and

$$\phi_7(\boldsymbol{x}_e, \alpha_{12}) = (\alpha_{12} - \underline{\alpha}_{12})(\alpha_{12} - \bar{\alpha}_{12}),$$

together define the estimate of $\mathscr{D} \times \mathscr{P} \subseteq \Phi = \{(\boldsymbol{x}_e, \alpha_{12}) \mid \phi_i \le 0 \ \forall i = 1,\ldots,7\}$. ◊

We implemented this program in SOSTOOLS, taking $\underline{\alpha}_{12} = 1$ and $\bar{\alpha}_{12} = 10$, and searching over $\deg(V) = 4$ and $\deg(\sigma_i) = 4$ for $i = 1,\ldots,7$. The resulting semi-definite program is in 24,256 dimensions, and takes just under 3 minutes to complete. The optimal value of $V(1,2,-1,1,2,-1)$ is equal to 0.0817. Hence, independently of the value of the parameter $\alpha_{12} \in [1,10]$, the error in the trajectory of the concentration of E incurred by doubling the rate of the reaction $P_1 \to E_1$ from $\alpha_1 = 1$ to $\tilde{\alpha}_1 = 2$ is bounded above such that $\|e - \tilde{e}\|^2 \le 0.0817$. Figure 3.3 compares this bound to 100 simulated estimates of the error for varying values of α_{12} in the allowed range. Thus, a single SOS program can certify an upper bound on the worst-case error for a widely-varying parameter value. As in the previous example, this result can be also extended to consider the worst-case error over varying initial conditions.

sQSSA Reduction	deg$(V) = 4$	deg$(V) = 6$	deg$(V) = 8$
$\boldsymbol{\pi}_1$	0.0094 (1.02 s)	0.0084 (2.22 s)	0.0075 (7.13 s)
$\boldsymbol{\pi}_2$	0.0174 (1.25 s)	0.0155 (2.72 s)	0.0153 (7.37 s)
$\boldsymbol{\pi}_3$	0.0464 (1.00 s)	0.0464 (2.26 s)	0.0464 (7.86 s)
$\boldsymbol{\pi}_4$	0.0889 (1.14 s)	0.0891 (2.51 s)	0.0890 (7.47 s)
Dimension	781	1685	4265

rQSSA Reduction	deg$(V) = 4$	deg$(V) = 6$	deg$(V) = 8$
$\boldsymbol{\pi}_1$	9.3817 (2.16 s)	9.3707 (4.00 s)	9.3741 (10.95 s)
$\boldsymbol{\pi}_2$	7.0400 (2.13 s)	7.0273 (3.92 s)	7.0283 (11.21 s)
$\boldsymbol{\pi}_3$	0.3526 (2.20 s)	0.3525 (3.32 s)	0.3525 (10.38 s)
$\boldsymbol{\pi}_4$	0.1643 (2.01 s)	0.1640 (3.33 s)	0.1640 (10.79 s)
Dimension	1405	2870	7081

tQSSA Reduction	deg$(V) = 4$	deg$(V) = 6$	deg$(V) = 8$
$\boldsymbol{\pi}_1$	0.0014 (1.17 s)	0.0012 (3.36 s)	0.0016 (9.26 s)
$\boldsymbol{\pi}_2$	0.0018 (1.80 s)	0.0016 (3.17 s)	0.0018 (9.27 s)
$\boldsymbol{\pi}_3$	0.0132 (2.38 s)	0.0118 (3.20 s)	0.0118 (10.39 s)
$\boldsymbol{\pi}_4$	0.0088 (1.92 s)	0.0082 (3.22 s)	0.0083 (9.81 s)
Dimension	1405	2597	6640

Table 3.2: Upper bounds $V(\boldsymbol{x}_e(0))$ on square of error $\|x - \tilde{x}\|^2$ incurred applying sQSSA, rQSSA, and tQSSA model reduction, calculated using SOSTOOLS according to Programs 3.6, 3.7, and 3.8 respectively, with deg$(\phi_i) = 4$. The code used to produce the sQSSA results can be found in Appendix A.

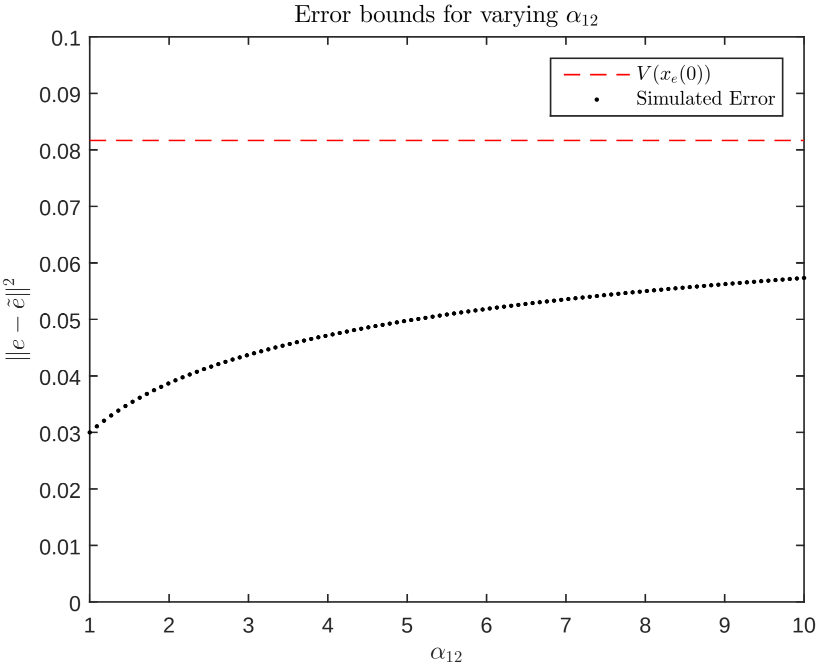


Figure 3.3: This plot compares 100 simulations of the error $\|e - \tilde{e}\|^2$, for the varying parameter values $\alpha_{12} \in [1, 10]$, with the upper bound on the worst case error given by $V\big(\boldsymbol{x}_e(0)\big) = 0.0817$ found by Program 3.11. Here $\alpha_1 = \alpha_2 = \tilde{\alpha}_2 = 1$, the perturbed parameter $\tilde{\alpha}_1 = 2$, and the initial condition parameters are $A_1 = 1$, and $A_2 = 2$.

3.4 Discussion

In this chapter we have described how to employ the SOS implementation of dissipativity analysis to estimate the response of an autonomous system's output trajectory to two types of perturbation: the structural perturbation of model order reduction, and parametric perturbation. Each of these cases was illustrated with an example. In the first of these, we used SOS programming to bound the error incurred by three alternative methods for the quasi-steady state approximation of enzyme kinetics. Each of these candidate QSSA methods has been justified analytically as being more or less valid in particular parameter regions [28, 205]. However, given a particular vector of parameters and a range of initial conditions, we can instead quantify a reduction's validity computationally in terms of a guarantee on the worst-case output error incurred. Our method gives an *a priori* estimate of the worst case error, thereby allowing us to evaluate the validity of each type of QSSA without requiring exhaustive simulation.

The calculation of the worst-case error across a set of possible initial conditions $\mathscr{X}_0$ and/or parameter values $\mathscr{P}$ is a key benefit of the SOS technique. Rather than producing a large number of simulations of a system, where each simulation corresponds to a distinct choice of $\left(\boldsymbol{x}_e(0), \boldsymbol{\pi}, \tilde{\boldsymbol{\pi}}\right)$ in $\mathscr{X}_0 \times \mathscr{P}$, a single SOS program can produce a constant B^2 that is a guaranteed upper bound on the error in output trajectories across all $\left(\boldsymbol{x}_e(0), \boldsymbol{\pi}, \tilde{\boldsymbol{\pi}}\right)$ in $\mathscr{X}_0 \times \mathscr{P}$. Thus, for this technique to be beneficial, running Program 3.4 should be faster than carrying out the set of distinct simulations required by a sufficiently fine gridding of $\mathscr{X}_0 \times \mathscr{P}$. Section 3.4.3 discusses the issue of computation speed in more detail.

In traditional QSSA reduction analysis, as described in Section 3.2.1, the reduction is based on the existence of a small parameter, so that the dynamics can be expressed in terms of the timescale-separated dynamics in (1.5). Theorem 1.1 states that the solutions of the full and reduced ODEs converge as the small parameter μ approaches zero [108]. Thus, if μ is 'small enough', the reduction is a close approximation. However, the 'small enough' criterion can only be evaluated relative to the scales of the other parameters. We saw in Section 3.2.2 that for the same small value of E_0/X_0, different parameter values gave different errors. Hence a more quantitative approach, which evaluates the error in the trajectory of a particular output of importance, is going to supply a more precise answer to whether Tikhonov's Theorem applies or not in any given parameter regime.

Section 3.4.1 discusses the potential application of SOS error bounding for the development of an automated, algorithmic approach to model order reduction based on error minimisation. After this, Sections 3.4.2 and 3.4.3 consider the issues of conservatism and computational burden associated with the model order reduction of large-scale, uncertain autonomous systems.

3.4.1 Error-based model reduction algorithm

Suppose we intend to reduce a large-scale ODE model Γ of a biomolecular reaction network, of dimension N, by constructing a reduced model $\tilde{\Gamma}$ of dimension $\tilde{N} \ll N$. Recall the derivation of singularly-perturbed dynamics given in Section 1.2.2. The key question is how to write the dynamics of Γ as (1.5), and hence define $\tilde{\Gamma}$ by (1.6).

Consider a partition of the state vector $\boldsymbol{x}^T = (\tilde{\boldsymbol{x}}^T, \hat{\boldsymbol{x}}^T)$ where $\tilde{\boldsymbol{x}} \in \mathbb{R}^{\tilde{N}}$. By partitioning the rows of the stoichiometric matrix into $\boldsymbol{S}^T = [\tilde{\boldsymbol{S}}^T, \hat{\boldsymbol{S}}^T]$ conformally with $(\tilde{\boldsymbol{x}}^T, \hat{\boldsymbol{x}}^T)$, the reduced dynamics are

$$\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{S}}\boldsymbol{v}\big(\tilde{\boldsymbol{x}}, \boldsymbol{\phi}(\tilde{\boldsymbol{x}})\big), \tag{3.24a}$$

where $\boldsymbol{\phi}(\tilde{\boldsymbol{x}})$ is the solution to

$$\boldsymbol{0} = \hat{\boldsymbol{S}}\boldsymbol{v}\big(\tilde{\boldsymbol{x}}, \boldsymbol{\phi}(\tilde{\boldsymbol{x}})\big), \tag{3.24b}$$

and takes values in $\mathbb{R}^{N-\tilde{N}}$. We can then compare the output $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x})$ of the large-scale system with the output $\tilde{\boldsymbol{y}} = \boldsymbol{h}\big(\tilde{\boldsymbol{x}}, \boldsymbol{\phi}(\tilde{\boldsymbol{x}})\big)$. Assume that the initial conditions $\tilde{\boldsymbol{x}}(0)$ also correspond to the partition $\boldsymbol{x}(0)^T = \big(\tilde{\boldsymbol{x}}(0)^T, \hat{\boldsymbol{x}}(0)^T\big)$.

We define the task of a model reduction algorithm as the optimal selection of which $\tilde{N}$ components of $\boldsymbol{x}$ to include in $\tilde{\boldsymbol{x}}$. There are $\binom{N}{\tilde{N}}$ possible selections, which rapidly increases for large-scale values of N. Thus, it can quickly become impractical to choose the best components to include by computationally evaluating the error incurred through all possible selections. Note also that this approach is fairly restrictive; in tQSSA, for example, we do not select a single state from (x, c) to keep in $\tilde{\boldsymbol{x}}$, but instead use $\tilde{\boldsymbol{x}} = (\xi)$ for a linear combination $\xi = x + c$ of the original states. Nevertheless, for now we consider the limited case of selecting x_i from $\boldsymbol{x}$ to use in $\tilde{\boldsymbol{x}}$.

A more promising approach to designing a model order reduction algorithm may be to use a greedy algorithm that sequentially reduces a large-scale system by discarding only one state at each step [8] . The state discarded is the state where we can guarantee the worst-case error to be smallest, applying the SOS error bounding technique introduced in this chapter. If there is a pre-determined error tolerance $E^2 > 0$, this algorithm can also discard as many states as possible while guaranteeing the eventual error bound $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 \leq E^2$. Consider the algorithm adapted from [8] given below. Its inputs are the large-scale system Γ with dynamics $\dot{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x})$, and an error threshold E^2. Its outputs are the reduced system $\tilde{\Gamma}$ of dimension $\tilde{N}$ such that $\max \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 \leq E^2$.

Algorithm 3.12 (Model Reduction)

```
  1   Set $n := N$, $r := 1$
  2   Set $\boldsymbol{S}^{(r)} := \boldsymbol{S}$, $\boldsymbol{v}^{(r)} := \boldsymbol{v}$
  3   For $i = 1 : n$
3.1       Define $\Gamma^{(r)}$ by dynamics $\dot{\boldsymbol{x}} = \boldsymbol{S}^{(r)}\boldsymbol{v}^{(r)}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{R}^n$.
3.2       Define $\hat{\boldsymbol{S}}_i^{(r)} = (\boldsymbol{S}_i^{(r)})$ and $\tilde{\boldsymbol{S}}_i^{(r)} = (\boldsymbol{S}_k^{(r)})_{k \neq i}$ for rows $\boldsymbol{S}_k^{(r)}$ of $\boldsymbol{S}^{(r)}$
3.3       Define $\tilde{\Gamma}_i^{(r)}$ by dynamics (3.24) for stoichiometries $\tilde{\boldsymbol{S}}_i^{(r)}$ and $\hat{\boldsymbol{S}}_i^{(r)}$
3.4       Find an upper bound $\epsilon_i^{(r)}$ on the worst-case error between $\Gamma^{(r)}$ and $\tilde{\Gamma}_i^{(r)}$
3.5       Loop
  4   If $\min_i \epsilon_i^{(r)} \leq E^2$
4.1       Set $j := \arg\min_i \epsilon_i^{(r)}$
4.2       Set $\boldsymbol{S}^{(r+1)} := \tilde{\boldsymbol{S}}_j^{(r)}$
4.3       Define $\boldsymbol{\phi} : \mathbb{R}^{n-1} \mapsto \mathbb{R}$ to solve $\boldsymbol{0} = \hat{\boldsymbol{S}}_j^{(r)}\boldsymbol{v}^{(r)}\big(\boldsymbol{x}, \boldsymbol{\phi}(\boldsymbol{x})\big)$
4.4       Set $\boldsymbol{v}^{(r+1)} : \mathbb{R}^{n-1} \mapsto \mathbb{R}$ with $\boldsymbol{v}^{(r+1)}(\boldsymbol{x}) := \boldsymbol{v}^{(r)}\big(\boldsymbol{x}, \boldsymbol{\phi}(\boldsymbol{x})\big)$.
4.5       Set $E^2 := E^2 - \epsilon_j^{(r)} \geq 0$
4.6       Set $n := n - 1$, $r := r + 1$
4.7       Go to 3
  5   Else return $\tilde{\Gamma} = \Gamma^{(r)}$ of dimension $\tilde{N} = n$
  6   End
```

The greediness of this algorithm is a result of the assumption that the best way to discard k states is to sequentially remove the k states whose removal at each stage gives minimal error. However, it is not immediately clear that this heuristic is optimal, since states may interact in such a way that the removal of two or more at once incurs far less error than that which can be estimated by their sequential removal. Furthermore, this entire approach is based on discarding a state according to the QSSA paradigm (3.24). There may be many other ways to discard a state; for example, by assuming it is much *slower* than the other states and thereby setting it to be constant. Finally, as mentioned above, using linear or nonlinear combinations of states in the reduced model may result in much less error. These issues are likely to inform the potential directions of future research into more sophisticated algorithmic approaches to model order reduction based on error minimisation.

3.4.2 Conservatism and slack

Figures 3.2 and 3.3 show how the error bound $V\big(\boldsymbol{x}_e(0)\big)$ found by a single SOS program compares to a large number of simulations across an entire range of initial conditions and uncertain parameter values. In each of these cases, it is clear there is some conservatism in the bound, depending on the specific SOS program used to construct V. The following result shows that the source of this gap is the slack caused by constraining the state and parameter space $\mathscr{D} \times \mathscr{P} \subseteq \Phi$ according to the technique in Section 2.3.

Proposition 3.13. *Consider V and σ_i given by Program 3.3. Define the* slack *$h(\boldsymbol{x}_e)$ as a function of the trajectory $\boldsymbol{x}_e(t)$ by*

$$h(\boldsymbol{x}_e) = -\frac{\sum_{i=1}^{R} \sigma_i(\boldsymbol{x}_e)\phi_i(\boldsymbol{x}_e)}{\psi(\boldsymbol{x}_e)} \geq 0, \tag{3.25}$$

which is non-negative for $\boldsymbol{x}_e \in \mathscr{D} \subseteq \Phi = \big\{\boldsymbol{x}_e \mid \phi_i(\boldsymbol{x}_e) \leq 0\big\}$. For a fixed initial condition $\boldsymbol{x}_e(0) \in \mathscr{X}_0$, the gap between the actual error $\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|^2$ and the upper bound on the error $V\big(\boldsymbol{x}_e(0)\big)$ is bounded below such that

$$V\big(\boldsymbol{x}_e(0)\big) - \|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|^2 \geq \int_0^\infty h\big(\boldsymbol{x}_e(t)\big)\,\mathrm{d}t = H\big(\boldsymbol{x}_e(0)\big) \geq 0.$$

Proof. The SOS inequality (3.5c) implies that

$$-\big[(\boldsymbol{y}-\tilde{\boldsymbol{y}})^2 + \dot{V}(\boldsymbol{x}_e)\big] \geq h(\boldsymbol{x}_e).$$

Integrating this inequality over $t \in [0,\infty)$ gives the required result. □

In the case of the sQSSA example of Figure 3.2, each of the three storage functions $V^\star$, V, and V_{tr} depicted satisfies (3.14c), and has an associated slack $h(\boldsymbol{x}_e)$ and resulting gap $H\big(\boldsymbol{x}_e(0)\big)$.

In the case of fixed initial conditions $\boldsymbol{x}_e(0)$, minimising $V\big(\boldsymbol{x}_e(0)\big)$ will have the effect of minimising $H\big(\boldsymbol{x}_e(0)\big)$. Hence, the storage function V found through Program 3.6 and shown in Figure 3.2 should ensure that $H(X_0, 0, X_0)$ is minimal. Indeed, Figure 3.2 shows almost no gap between the simulated error $\|x-\tilde{x}\|^2$ and the upper bound $V(X_0, 0, X_0)$, which means that $H\big(\boldsymbol{x}_e(0)\big)|_{x_0=X_0} \approx 0$. This in turn implies that $h\big(\boldsymbol{x}_e(t)\big) \approx 0$ along the trajectory initialised from $x_0 = X_0$. The red curve in Figure 3.4 plots the trajectory of $h\big(\boldsymbol{x}_e(t)\big)$, for the multipliers σ_i given by Program 3.6. Clearly $h \approx 0$ for all t. Thus, the SOS program has optimised the multipliers σ_i such that the

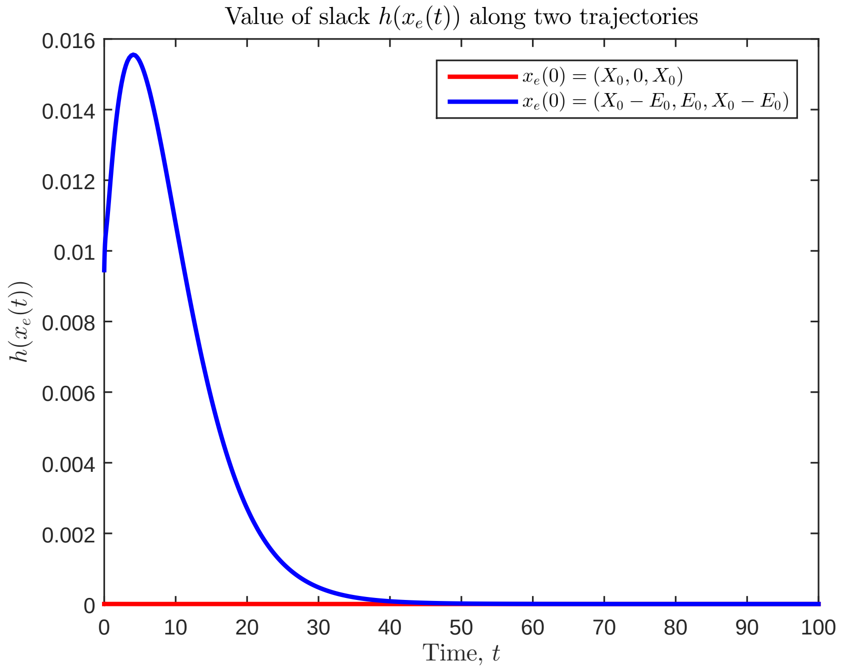


Figure 3.4: For the storage function V and multipliers σ_i found through Program 3.6 with parameter values $\boldsymbol{\pi}_1$, we track the slack function $h\big(\boldsymbol{x}_e(t)\big)$ along two trajectories with distinct initial conditions. For $\boldsymbol{x}_e(0) = (X_0, 0, X_0)$, the slack is approximately zero and $V\big(\boldsymbol{x}_e(0)\big)$ is a tight upper bound on $\|x - \tilde{x}\|^2$. When $\boldsymbol{x}_e(0) = (X_0 - E_0, E_0, X_0 - E_0)$, the slack has positive integral $H\big(\boldsymbol{x}_e(0)\big)$ and there is therefore a significant gap between the estimate and the real error value.

trajectory of the error system's state $\boldsymbol{x}_e$ takes values $\boldsymbol{x}_e(t)$ in the kernel $\big\{\boldsymbol{x}_e \in \mathscr{D} \mid h(\boldsymbol{x}_e) = 0\big\}$ of the slack function h.

However, consider the distinct trajectory with initial condition parameter $x_0 = X_0 - E_0$. The slack function $h\big(\boldsymbol{x}_e(t)\big)$ associated with V is no longer close to zero along this trajectory, as shown by the blue curve in Figure 3.4. Hence the total slack $H\big(\boldsymbol{x}_e(0)\big)$ is significantly larger than zero, causing $V\big(\boldsymbol{x}_e(0)\big)$ to be a much more conservative bound for this second initial condition.

In the case of parameter uncertainty, the slack $h(\boldsymbol{x}_e, \boldsymbol{\pi})$ can be defined similarly to (3.25), with additional $\boldsymbol{\pi}$ dependence. The storage function $V(\boldsymbol{x}_e)$ used to create Figure 3.3 has been constructed by Program 3.11 to minimise $V\big(\boldsymbol{x}_e(0)\big)$ for a given, fixed initial condition. However, there is a significant gap between this bound and the simulated errors for each value of α_{12}. In this case, we can conclude that the total slack $H\big(\boldsymbol{x}_e(0), \alpha_{12}\big)$ defined by the SOS multipliers σ_i given by Program 3.11 is always positive for each value of $\alpha_{12} \in [1, 10]$. For example, when $\alpha_{12} = 5$, the value of $\phi_7(\boldsymbol{x}_e, 5) = -20$ over the entire trajectory. Hence $H\big(\boldsymbol{x}_e(0), 5\big) \geq 20 \int_0^\infty \sigma_7(\boldsymbol{x}_e, 5)\, \mathrm{d}t$. For a non-zero multiplier σ_7, this necessarily results in a positive value for the slack $H\big(\boldsymbol{x}_e(0), 5\big)$.

Clearly, to improve the quality of the upper error bounds found through SOS programming, we must attempt to reduce the maximum size of the slack $H\big(\boldsymbol{x}_e(0), \boldsymbol{\pi}\big)$ over the set $\mathscr{X}_0 \times \mathscr{P}$ of initial conditions and uncertain parameters. The slack $H\big(\boldsymbol{x}_e(0), \boldsymbol{\pi}\big)$ is zero, and hence the estimate $V\big(\boldsymbol{x}_e(0)\big)$ is at its tightest, if the trajectory of the error system state takes values $\boldsymbol{x}_e(t)$ in the kernel of $h(\boldsymbol{x}_e, \boldsymbol{\pi})$. An important area of further work will be to exploit an analysis of the form of h to choose the multiplier forms σ_i and τ_j, the domain-constraining polynomials ϕ_i and θ_j, and a more appropriate objective function, so that the optimisation programs given by Program 3.3 or

Program 3.4 can find the tightest possible error bound $V\big(\boldsymbol{x}_e(0)\big)$ or B^2.

3.4.3 Scaling

Consider the following BRN, combining the two examples in Sections 3.2 and 3.3 and an additional reaction to give $M = 7$ reactions on $N = 10$ reactants, such that

$$P_i \xrightarrow{\alpha_i} E_i, \tag{3.26a}$$

$$E_1 + E_2 \xrightarrow{\alpha_{12}} E, \tag{3.26b}$$

$$X + E \xrightleftharpoons[\beta_{-1}]{\beta_1} C \xrightarrow{\beta_2} Y + E, \tag{3.26c}$$

$$Y + K \xrightarrow{\beta_3} Z, \tag{3.26d}$$

for $i = 1, 2$. Here, the enzyme E is constructed from precursors P_1 and P_2. The enzyme is used to catalyse the production of the product protein Y from substrate protein X. Finally, the product Y is activated by K to the active form Z. As before, the concentrations of these species are denoted by lowercase letters. The dynamics of this system are written in the form of (1.2) as

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_1 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e} \\ \dot{x} \\ \dot{c} \\ \dot{y} \\ \dot{k} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 p_1 \\ \alpha_2 p_2 \\ \alpha_{12} e_1 e_2 \\ \beta_1 x e \\ \beta_{-1} c \\ \beta_2 c \\ \beta_3 y k \end{bmatrix}. \tag{3.27}$$

We assume that the initial conditions are fixed such that $p_1(0) = A_1$, $p_2(0) = A_2$, $x(0) = X_0$, $y(0) = Y_0$, and $k(0) = K_0$, with all other initial conditions $e_1(0) = e_2(0) = e(0) = c(0) = z(0) = 0$. As in the previous examples, a number of conservation relations apply, which eventually allow these dynamics to be written as a 6 dimensional ODE system.

Suppose we wish to bound the response of the trajectory of z to the parametric perturbation of $\alpha_1 = 1$ into $\tilde{\alpha}_1 = 2$. A SOS program can be formulated that, when solved, produces a storage function V to bound the response of the system output to this perturbation. The state $\boldsymbol{x}_e = (\boldsymbol{x}, \tilde{\boldsymbol{x}})$ of the error system takes values in 12-dimensional space. A SOS polynomial $V \in \Sigma$ of degree 6 in 12 variables has 206,116 associated decision variables. It takes SOSTOOLS 50 minutes simply to initialise this many decision variables into memory. Computation on a semi-definite program of this size is, clearly, not possible.

This example illustrates the fact that, as mentioned in Section 2.4, a major issue with the use of SOS programming for the dissipativity analysis of ODE systems is the rapid increase in problem dimension as the number of state components grows. In general, the error system associated with perturbing an N-dimensional system into an $\tilde{N}$-dimensional system is of dimension

$N + \tilde{N}$. The typical motivation for model reduction is that N, and hence $N + \tilde{N}$, is impractically large. This means that, when applying the SOS methods described in this chapter to the model reduction of large-scale systems, the resulting SOS program is likely to be of unmanageably high dimension. The effect of high dimensionality is exacerbated in the presence of parametric uncertainty. Since the uncertain parameters require additional auxiliary variables to be introduced to the SOS program, they add extra state variables to the problem, compounding the effect of high dimensionality.

When applied to many systems analysis tasks, the decomposition of large-scale systems is a promising strategy for overcoming the problems of high dimensionality and computational overload. A natural approach to the dimensionality problem identified above is to decompose the SOS programming approach to dissipativity analysis by exploiting the system's structure. In the remainder of this thesis, we seek to develop methods for the structural analysis and decomposition of BRNs. The application of the decomposition techniques to the problems introduced in this chapter will be discussed in Chapter 7. There we will show that dissipativity-based error estimation of a large-scale BRN through SOS programming can be made much more tractable by exploiting its layered structure, introduced in the next chapter.

Chapter 4

Layered Reaction Networks

As discussed in Section 1.3, model decomposition is a common technique used to simplify the simulation, analysis, and control of large-scale systems. This chapter introduces a new framework for the decomposition of large-scale BRN models into interconnected subsystems, which we term *layers*. This approach is distinct from established, modular decomposition methods in that each subsystem may contribute to the dynamics of all reactants. The input–output behaviour of each layer is shown to be independent of the other layers in the network. After defining a layered decomposition, we begin to consider how to uncover the layered structure of a BRN, paying particular attention to ensuring that the layers of a stable, large-scale network are stable subsystems. The task of detecting a layered structure within a BRN will provide the motivation for Chapter 5 and Chapter 6.

4.1 Modular Networks

Decomposition is a typical approach used in Systems Biology for overcoming the complexity of BRN models by exploiting their structure. The concept of *modules* forms a widely-accepted decomposition framework. In modular decompositions, subsystems are defined by simultaneously partitioning the sets $\mathscr{S} = \{X_1,\ldots,X_N\}$ and $\mathscr{R} = \{R_1,\ldots,R_M\}$ of species X_i and reactions R_j respectively into L subsets $\mathscr{S}_l \subset \mathscr{S}$ and $\mathscr{R}_l \subset \mathscr{R}$. For each $l = 1,\ldots,L$, the pair $(\mathscr{S}_l,\mathscr{R}_l)$ together define a module, where the concentrations $x_j(t)$ of species $X_j \notin \mathscr{S}_l$ are external inputs into module l [167]. There are two problems associated with modular decomposition. The first is to identify the correspondence between modules and their biological interpretation. The second is illustrated by the following example, which demonstrates that modules may not have well-defined dynamics because they do not directly model interconnection effects.

Example 4.1 Consider the following gene transcription cascade

$$U + G_X \underset{k_{X,OFF}}{\overset{k_{X,ON}}{\rightleftharpoons}} (U : G_X) \xrightarrow{k_X} X \xrightarrow{\delta_X}, \tag{4.1a}$$

$$X + G_Y \underset{k_{Y,OFF}}{\overset{k_{Y,ON}}{\rightleftharpoons}} (X : G_Y) \xrightarrow{k_Y} Y \xrightarrow{\delta_Y}, \tag{4.1b}$$

describing how an externally-controlled protein U acts as a promoter of the gene G_X, the product of which is the promoter X of the downstream gene G_Y, the product of which is Y. The time-varying concentration $u(t)$ of U is an input to this BRN, and the concentration $y(t)$ of Y is the output. The corresponding species and reaction sets are

$$\begin{aligned}
\mathscr{S} &= \{G_X, (U : G_X), X, G_Y, (X : G_Y), Y\},\\
\mathscr{R} &= \left\{R_{k_{X,ON}}, R_{k_{X,OFF}}, R_{k_X}, R_{\delta_X}, R_{k_{Y,ON}}, R_{k_{Y,OFF}}, R_{k_Y}, R_{\delta_Y}\right\},
\end{aligned}$$

where U is considered to be external to the network, and is therefore not included in $\mathscr{S}$.

Each line of (4.1) suggests a natural modular decomposition of this network into $L = 2$ gene transcription modules, defined by the partition $\mathscr{S}_1 = \{G_X, (U : G_X), X\}$ with $\mathscr{S}_2 = \mathscr{S} \setminus \mathscr{S}_1$, and $\mathscr{R}_1 = \{R_{k_{X,ON}}, R_{k_{X,OFF}}, R_{k_X}, R_{\delta_X}\}$ with $\mathscr{R}_2 = \mathscr{R} \setminus \mathscr{R}_1$. The first module is given by

$$G_X \underset{k_{X,OFF}}{\overset{k_{X,ON}u(t)}{\rightleftharpoons}} (U : G_X) \xrightarrow{k_X} X \xrightarrow{\delta_X} \tag{4.2a}$$

with input $u(t)$ representing the concentration of an external variable. The output $x(t)$ of module (4.2a) is the external input to the second module

$$G_Y \underset{k_{Y,OFF}}{\overset{k_{Y,ON}x(t)}{\rightleftharpoons}} (X : G_Y) \xrightarrow{k_Y} Y \xrightarrow{\delta_Y} \tag{4.2b}$$

with output $y(t)$.

Assume for simplicity that the concentrations of G_X and G_Y are constant, and incorporate their proportional effect into the new parameters $\hat{k}_{X,ON} = k_{X,ON}G_X$ and $\hat{k}_{Y,ON} = k_{Y,ON}G_Y$. The isolated dynamics of $(U : G_X)$ and X in module (4.2a) are

$$\dot{c}_X(t) = \hat{k}_{X,ON}u(t) - k_{X,OFF}c_X(t) - k_X c_X(t), \tag{4.3a}$$

$$\dot{x}(t) = k_X c_X(t) - \delta_X x(t), \tag{4.3b}$$

where c_X represents the concentration of bound gene $(U : G_X)$.

When the modules are interconnected together, such that $x(t)$ forms the input to the downstream module (4.2b), the downstream module has dynamics

$$\dot{c}_Y(t) = \hat{k}_{Y,ON}x(t) - k_{X,OFF}c_Y(t) - k_X c_Y(t), \tag{4.4a}$$

$$\dot{y}(t) = k_Y c_Y(t) - \delta_Y y(t). \tag{4.4b}$$

However, the trajectory of $x(t)$ in the interconnected system is not the same as the trajectory given by the isolated dynamics (4.3b). If the interconnected dynamics of $x(t)$ are directly derived from the model (4.1), then the correct version of (4.3b) is

$$\dot{x}(t) = k_X c_X(t) - \delta_X x(t) + \left[k_{Y,OFF}c_Y(t) - \hat{k}_{Y,ON}x(t)\right]. \tag{4.5}$$

The mechanism of the module interconnection is such that X reversibly binds with G_Y to form the complex $(X : G_Y)$. Thus, although the modular formulation appears to imply a cascade from the X-module to the Y-module, the dynamics of x upon interconnection change to depend on the downstream variable $c_Y(t)$ and downstream parameters $\hat{k}_{Y,ON}$ and $k_{Y,OFF}$, as shown in (4.5). Hence the module dynamics need to be remodelled when interconnected with another module;

the module is not well-defined. This example illustrates the phenomenon of *retroactivity* [55, 56, 83]. ♢

The *modularity* of a given reaction network is a concept which describes how well it decomposes into modules. It is negated by nonlinear interaction effects between modules such as those caused by retroactivity, introduced in Example 4.1 above, together with crosstalk in signalling pathways [21, 126, 171], cell burden [50, 185], and so on. These effects cause a module to behave differently when integrated with a larger system. In the following section we will present a novel framework for decomposing a BRN, where the input–output behaviour of each resulting subsystem is always the same, regardless of which other subsystems it is connected with.

4.2 Decomposing Layered Networks

The modular decomposition of a BRN simultaneously partitions the reaction set $\mathscr{R}$ and species set $\mathscr{S}$ into subsets, each pair of which defines a module. Hence (without loss of generality) the state vector $\boldsymbol{x}^T = [\boldsymbol{x}_1^T, \dots, \boldsymbol{x}_L^T]$ is partitioned into L components, and the dynamics of each component $\boldsymbol{x}_l$ is given by module l. Our approach is to instead replicate the state vector $\boldsymbol{x} \in \mathbb{R}^N$ into multiple state vectors $\boldsymbol{x}^l \in \mathbb{R}^N$ for $l = 1, \dots, L$, each of which is the state of a subsystem, called a layer, defined as follows.

Definition 4.2 Consider the BRN model (1.2) with initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Suppose that the stoichiometric matrix $\boldsymbol{S}$ can be written as a sum of $L \in \mathbb{N}$ matrices $\boldsymbol{S}^l$ for $l = 1, \dots, L$ such that $\boldsymbol{S} = \boldsymbol{S}^1 + \boldsymbol{S}^2 + \dots + \boldsymbol{S}^L$. Define the L layered states $\boldsymbol{x}^l$ taking values $\boldsymbol{x}^l(t) \in \mathbb{R}^N$ with dynamics

$$\dot{\boldsymbol{x}}^l(t) = \boldsymbol{S}^l \boldsymbol{v}\big(\boldsymbol{x}^1(t) + \boldsymbol{x}^2(t) + \dots + \boldsymbol{x}^L(t)\big) \tag{4.6}$$

for each $l = 1, 2, \dots, L$. The initial conditions of each layered state $\boldsymbol{x}^l$ are $\boldsymbol{x}^l(0) = \boldsymbol{x}_0^l$, for arbitrary vectors $\boldsymbol{x}_0^l \in \mathbb{R}^N$ chosen to satisfy the condition $\boldsymbol{x}_0 = \sum_{l=1}^L \boldsymbol{x}_0^l$. Note that the layer index is in superscript to distinguish it from the module index in subscript. ♢

Proposition 4.3. *The sum of the layers' states $\boldsymbol{X}(t) = \sum_{l=1}^L \boldsymbol{x}^l(t)$ follows the same trajectory as the state $\boldsymbol{x}$ of the original system* (1.2).

Proof. The dynamics (4.6) can be written $\dot{\boldsymbol{x}}^l = \boldsymbol{S}^l \boldsymbol{v}\big(\boldsymbol{X}(t)\big)$. Summing the dynamics for $l = 1, \dots, L$ gives $\dot{\boldsymbol{X}}(t) = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{X}(t))$, with initial conditions $\boldsymbol{X}(0) = \sum \boldsymbol{x}_0^l = \boldsymbol{x}_0$. Since $\boldsymbol{X}$ has the same dynamics and initial conditions as $\boldsymbol{x}$, they therefore have the same trajectory. □

Each layer's state vector $\boldsymbol{x}^l$ takes values $\boldsymbol{x}^l(t) \in \mathbb{R}^N$. The component values $x_i^l(t)$ of each layer's state quantify the contribution of layer l to the total concentration $x_i(t)$ of species X_i, for $i = 1, \dots, N$, at time t. While biochemical constraints mean that the total concentrations $x_i(t) \geq 0$ are non-negative, the values of the layered state components $x_i^l(t)$ may be negative.

In Definition 4.2, the initial conditions $\boldsymbol{x}_0$ can be distributed arbitrarily across the decomposition to the initial conditions $\boldsymbol{x}_0^l$ of each layer's state. It can be assumed that there is a 'zero' layer, or 'initial condition layer', with state $\boldsymbol{x}^0$, stoichiometric matrix $\boldsymbol{S}^0 = \boldsymbol{0}$, and initial conditions $\boldsymbol{x}_0^0 = \boldsymbol{x}_0$. This definition implies that $\boldsymbol{x}^0(t) \equiv \boldsymbol{x}_0$ for all t. By defining each of the other layers' initial conditions $\boldsymbol{x}_0^l = \boldsymbol{0}$ for $l = 1, \ldots, L$, then the layered dynamics (4.6) are re-written as

$$\dot{\boldsymbol{x}}^l(t) = \boldsymbol{S}^l \boldsymbol{v}\left(\boldsymbol{x}_0 + \sum_{l=1}^{L} \boldsymbol{x}^l(t)\right) \tag{4.7}$$

for $l = 1, \ldots, L$. The state is recovered by the sum $\boldsymbol{x}(t) = \boldsymbol{x}_0 + \sum_{l=1}^{L} \boldsymbol{x}^l(t)$.

Decomposing a system into a collection of overlapping subsystems has been previously investigated in the context of linear systems [18, 96] and nonlinear ecological networks [95]. In the latter, the state vectors of the subsystems are defined such that certain components take part in the dynamics of more than one subsystem. The dynamics and interactions of the subsystems are defined so that the components which take part in multiple subsystems have exactly the same trajectory in each. This is distinct from our strategy of letting $x_i^l(t) \neq x_i^k(t)$ in general. In the context of BRNs, there has also been recent work using overlapping subsystems in the context of model validation [115] and the composition of synthetic biomolecular components [135].

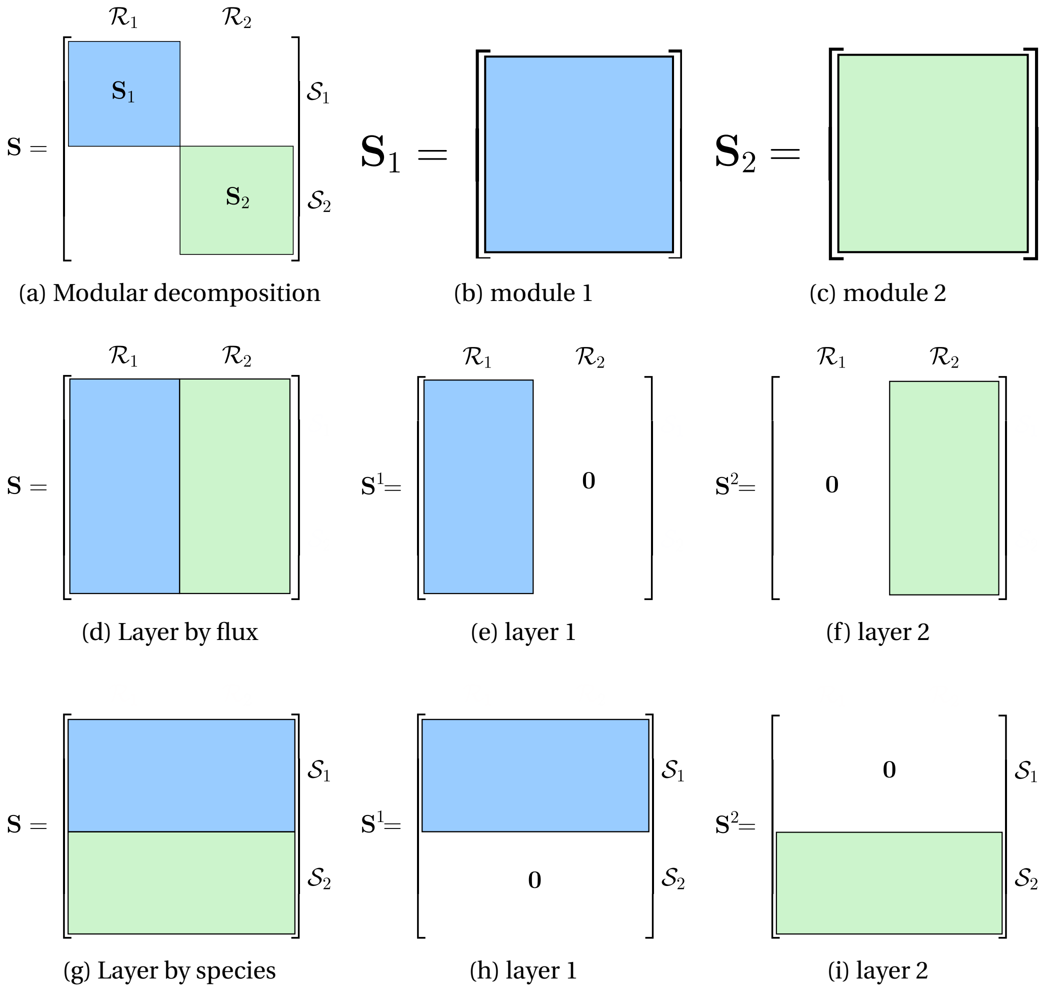
Each layer's dynamics (4.7) can also be written as

$$\dot{\boldsymbol{x}}^l(t) = \boldsymbol{S}^l \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{x}^l(t) + \boldsymbol{u}^l(t)\big), \tag{4.8}$$

where the external input $\boldsymbol{u}^l(t) = \sum_{k \neq l} \boldsymbol{x}^j(t)$ denotes the aggregated contribution of all the other layers. The dynamics of each $\boldsymbol{x}^l$, in general, depend on any of the other layers' states $\boldsymbol{x}^k$, $k \neq l$. Nevertheless, the input–output behaviour of each layer does not change in the context of the other layers. Although the trajectory of layer l will be affected by the presence of layer $k \neq l$, this is due to the input $\boldsymbol{u}^l$; the dynamics (4.8) remain unaffected. Hence, unlike modules, layers are well-defined subsystems.

To emphasise the distinction between layered and modular decomposition, recall that the latter is defined by a simultaneous partition of the species $\mathscr{S} = \{X_1, \ldots, X_N\}$ and reactions $\mathscr{R} = \{R_1, \ldots, R_M\}$ into L pairs of sets $(\mathscr{S}_l, \mathscr{R}_l)$ for $l = 1, \ldots, L$. In terms of the stoichiometry $\boldsymbol{S}$, each module can be modelled with dynamics $\dot{\boldsymbol{x}}_l = \boldsymbol{S}_l \boldsymbol{v}_l(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$ for the stoichiometric matrices $\boldsymbol{S}_l$ defined by deleting the rows i and columns j from $\boldsymbol{S}$ with $X_i \notin \mathscr{S}_l$ and $R_j \notin \mathscr{R}_l$. Without loss of generality, suppose that the species and reaction indices are ordered such that $\mathscr{S}_l = \{X_{1+i_{l-1}}, \ldots, X_{i_l}\}$ and $\mathscr{R}_l = \{R_{1+j_{l-1}}, \ldots, R_{j_l}\}$ for the increasing sequences $0 = i_0 < i_1 < \cdots < i_L = N$ and $0 = j_0 < j_1 < \cdots < j_L = M$. This definition of the modular stoichiometry $\boldsymbol{S}_l$ corresponds to taking the lth block along the diagonal of $\boldsymbol{S}$, as shown in the top row of Figure 4.1. The entries of $\boldsymbol{S}$ that are not contained in these blocks are neglected by the modular decomposition. The decomposition in [32], which the authors also refer to as a layered decomposition, is a modular decomposition of a block diagonal $\boldsymbol{S}$. In that case, the modular approach will not neglect any elements of $\boldsymbol{S}$; we will show below that the layered and modular decompositions of a block-diagonal $\boldsymbol{S}$ coincide.

Continued (Example 4.1, ctd) The distinction (for non-block diagonal $\boldsymbol{S}$) between layered and modular decompositions can be demonstrated using the BRN in Example 4.1 above. Assume for simplicity that the input $u(t) \equiv u$ is constant. The dynamics of the interconnected gene tran-



(a) Modular decomposition

(b) module 1

(c) module 2

(d) Layer by flux

(e) layer 1

(f) layer 2

(g) Layer by species

(h) layer 1

(i) layer 2

Figure 4.1: Schematic comparing modular decomposition (top row) with layering by flux (middle row) and layering by species (bottom row). The coloured areas denote where stoichiometric coefficients from $\boldsymbol{S}$ are used in each $\boldsymbol{S}^l$; white space denoted $\mathbf{0}$ corresponds to all-zero entries.

scription cascade (4.1) are

$$\begin{bmatrix} \dot{c}_X(t) \\ \dot{x}(t) \\ \dot{c}_Y(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} k_{X,ON}u \\ k_{X,OFF}c_X(t) \\ k_X c_X(t) \\ \delta_X x(t) \\ k_{Y,ON}x(t) \\ k_{Y,OFF}c_Y(t) \\ k_Y c_Y(t) \\ \delta_Y y(t) \end{bmatrix}. \tag{4.9}$$

The partitions $\{\mathscr{R}_1, \mathscr{R}_2\}$ and $\{\mathscr{S}_1, \mathscr{S}_2\}$ chosen in Section 4.1 correspond to taking the two 2×4 blocks along the diagonal of $\boldsymbol{S}$ as the modules' stoichiometric matrices. This neglects the non-zero elements $S_{2,5}$ and $S_{2,6}$. Retroactivity is therefore to be expected, since some of the interconnected system dynamics are not included in either of the modules.

We can now demonstrate the layered decomposition of this stoichiometry, according to Definition 4.2. We have not discussed how best to choose the set of layered stoichiometric matrices $\boldsymbol{S}^l$, subject to the constraint $\boldsymbol{S} = \sum_{l=1}^{L} \boldsymbol{S}^l$. Nevertheless, one possible choice for a decomposition of the system into $L = 2$ layers is given by

$$\boldsymbol{S}^1 + \boldsymbol{S}^2 = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \tag{4.10}$$

The dynamics of layer 1, written in full, are

$$\begin{bmatrix} \dot{c}_X^1(t) \\ \dot{x}^1(t) \\ \dot{c}_Y^1(t) \\ \dot{y}^1(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{k}_{X,ON}u \\ k_{X,OFF}\big(c_X^1(t) + c_X^2(t)\big) \\ k_X\big(c_X^1(t) + c_X^2(t)\big) \\ \delta_X\big(x^1(t) + x^2(t)\big) \\ \hat{k}_{Y,ON}\big(x^1(t) + x^2(t)\big) \\ k_{Y,OFF}\big(c_Y^1(t) + c_Y^2(t)\big) \\ k_Y\big(c_Y^1(t) + c_Y^2(t)\big) \\ \delta_Y\big(y^1(t) + y^2(t)\big) \end{bmatrix},$$

with a similar expression (replacing $\boldsymbol{S}^1$ with $\boldsymbol{S}^2$) for layer 2, and zero initial conditions for both. The zero rows of $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ imply that $\dot{c}_Y^1 = \dot{y}^1 = \dot{c}_X^2 = 0$. Since the initial values of these three

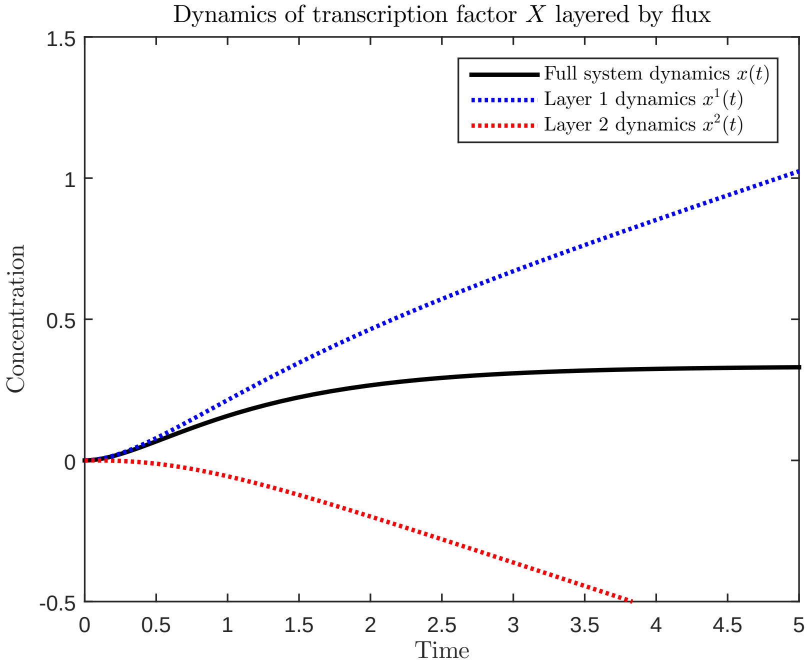


Figure 4.2: Three simulations of the trajectory $x(t)$, for parameters $k_i = \delta_i = 1$ and initial conditions $\boldsymbol{x}(0) = [0,0,0,0]^T$. The black solid line is the trajectory of $x(t)$ given by (4.9); the broken lines are the layered contributions $x^1(t)$ and $x^2(t)$ from (4.11) that each sum to give $x(t)$.

variables are zero, the layered dynamics simplify to

$$\begin{bmatrix} \dot{c}_X^1(t) \\ \dot{x}^1(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \hat{k}_{X,ON}\, u \\ k_{X,OFF}\, c_X^1(t) \\ k_X c_X^1(t) \\ \delta_X \big(x^1(t) + x^2(t)\big) \end{bmatrix}, \tag{4.11a}$$

$$\begin{bmatrix} \dot{x}^2(t) \\ \dot{c}_Y^2(t) \\ \dot{y}^2(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \hat{k}_{Y,ON}\big(x^1(t) + x^2(t)\big) \\ k_{Y,OFF}\, c_Y^2(t) \\ k_Y c_Y^2(t) \\ \delta_Y y^2(t) \end{bmatrix}, \tag{4.11b}$$

where the zero columns and rows from each $\boldsymbol{S}^l$ have been neglected. From (4.11) we can conclude that x^1 is an input to layer 2, and x^2 is an input to layer 1 (together with the external input signal u).

The trajectories of x^1, x^2, and $x = x^1 + x^2$ are depicted in Figure 4.2. The blue line shows that the effect of the load from layer 2 is to promote the production of X by layer 1. The red line represents the consumption by layer 2 of the supply of X provided by layer 1. The production and consumption of X by each layer reaches a balance as $t \to \infty$, so that $x(t) = x^1(t) + x^2(t) \to \bar{x}$ approaches steady state. We can show that layer 1 in isolation is stable, but the feedback interconnection destabilises layer 1. The stability properties of a layered decomposition are discussed in more detail in Section 4.3.

This example demonstrates a number of ways in which the layered and modular decomposition frameworks differ. In the layered decomposition the contribution x^2 of layer 2 to x is explicitly used as an input to layer 1. Hence the input–output behaviour of layer 1 remains the same, independently of whether it is isolated or integrated with layer 2. Unlike the modular case, it therefore does not need to be remodelled due to the other subsystems: the effect of its context is in the inputs c_x^2 and x^2. Note that, as stated above, while the input–output behaviour remains the same, this does not mean that the trajectory of x^1 is invariant to interconnection. Nevertheless, the effect of the interconnection on x can be modelled by the feedback loop and by summing the mutually unstable trajectories x^1 and x^2.

Furthermore, suppose that another reaction is added to the system, given by $\left(X \xrightarrow{\alpha} Z\right)$. Suppose that this reaction is put into a new layer. The x component of layer 3 is denoted x^3, with dynamics $\dot{x}^3 = -\alpha(x^1 + x^2 + x^3)$. Thus layer 3 is driven by the other two layers. Since it contributes the trajectory $x^3(t)$ with dynamics to the total concentration $x = x^1 + x^2 + x^3$, the trajectory $x^3(t)$ also forms an additional input to both layer 1 and 2. Nevertheless, neither layer 1 or layer 2 needs to be remodelled; the input–output behaviour of both layers remains the same, and only the trajectory of their inputs changes. This situation is distinct from the modular case, where the additional module would be modelled as $\left(\xrightarrow{\alpha x(t)} Z\right)$, driven by the trajectory $x(t)$ of X. However, the dynamics of x in the upstream module (4.3b) need to be further re-modelled by adding an additional retroactive term of $-[\alpha x(t)]$ to the right-hand side of (4.5). $\diamondsuit$

This section has introduced layering as a new approach to BRN decomposition that is complementary to established modular techniques. We have demonstrated that, in contrast to modular decomposition, layers are well-defined subsystems, the dynamics of which can be defined independently of the other layers in the system. However, as for any decomposition method, the fundamental problem of layered decomposition is how best to choose the subsystems. In the case of layered decompositions, we need to choose the set of matrices $\boldsymbol{S}^1, \ldots, \boldsymbol{S}^L$ which define each of the layers' dynamics (4.7) such that $\boldsymbol{S} = \sum_l \boldsymbol{S}^l$. The following section is concerned with a number of important properties of a layered BRN, which can be used to inform a decomposition strategy.

4.3 Layer Properties

This section assumes that a decomposition of the stoichiometric matrix $\boldsymbol{S} = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^L$ into L stoichiometric matrices is given, defining the layered dynamics (4.7).

4.3.1 Layer dimension

One of the key properties of any system decomposition is the dimension of the subsystems. A reasonable aim of any decomposition is that the subsystems are, in some sense, simpler than the large-scale system from which they are derived. In (4.7), each layer's state $\boldsymbol{x}^l$ takes values $\boldsymbol{x}^l(t) \in \mathbb{R}^N$ in a state space of the same dimension as the state space of the original system. Therefore it is not immediately clear that a layered decomposition defines simpler subsystems.

This problem is overcome by making a more sophisticated evaluation of each layer's dimension. The following results imply that the dimension of any BRN model $\dot{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x})$ is equal to the rank of the stoichiometric matrix $\boldsymbol{S}$. This property also devolves to the layers, so that each layer's dimension is equal to the rank of the layered stoichiometric matrix $\boldsymbol{S}^i$ in (4.7).

Lemma 4.4. *Consider any matrix $\boldsymbol{S} \in \mathbb{R}^{N\times M}$, and denote $r = \operatorname{rank}(\boldsymbol{S}) \le \min(N,M)$. Then there exist matrices $\boldsymbol{U} \in \mathbb{R}^{N\times r}$ and $\boldsymbol{C} \in \mathbb{R}^{r\times M}$ such that $\boldsymbol{S}$ can be expressed as the product $\boldsymbol{S} = \boldsymbol{UC}$.*

Proof. Consider the short form [15] of the SVD decomposition $\boldsymbol{S} = \boldsymbol{U\Sigma V}^T$, for $\boldsymbol{U} \in \mathbb{R}^{N\times r}$, $\boldsymbol{V} \in \mathbb{R}^{M\times r}$, and the square diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$ of singular values $\sigma_i > 0$. Writing $\boldsymbol{C} = \boldsymbol{\Sigma V}^T$ provides a decomposition of the required form $\boldsymbol{S} = \boldsymbol{UC}$. □

Note that the proof of Lemma 4.4 provides a specific $\boldsymbol{U}$ and $\boldsymbol{C}$ for each $\boldsymbol{S}$, but there are infinitely many such matrix pairs.

Proposition 4.5. *The ODE system* (1.2) *modelling a BRN can be expressed in terms of a state $\boldsymbol{\xi}$ taking values $\boldsymbol{\xi}(t) \in \mathbb{R}^r$, where the dimension of $\boldsymbol{\xi}$ is $r = \operatorname{rank}(\boldsymbol{S}) \le N$.*

Proof. Lemma 4.4 implies that the stoichiometric matrix can be written as the product $\boldsymbol{S} = \boldsymbol{UC}$. For the state vector $\boldsymbol{\xi}$ taking values $\boldsymbol{\xi}(t) \in \mathbb{R}^r$, define the dynamics

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{C}\boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{U}\boldsymbol{\xi}(t)\big) \tag{4.12}$$

with initial conditions $\boldsymbol{\xi}(0) = \boldsymbol{0}$, and output $\boldsymbol{x}(t) = \boldsymbol{x}_0 + \boldsymbol{U}\boldsymbol{\xi}(t)$ taking values $\boldsymbol{x}(t) \in \mathbb{R}^N$. Taking the derivative of $\boldsymbol{x}$ with respect to time gives $\dot{\boldsymbol{x}}(t) = \boldsymbol{UC}\boldsymbol{v}(\boldsymbol{x}(t))$, recovering the original system dynamics (1.2) with the same initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Thus (4.12) is a realisation of (1.2) in a state-space of dimension $r = \operatorname{rank}(\boldsymbol{S})$. □

Corollary 4.6. *The dimension of a layer's dynamics* (4.7) *is equal to $r^l = \operatorname{rank}(\boldsymbol{S}^l)$.*

Proof. Corollary 4.6 is proved in exactly the same way as Proposition 4.5, using the existence of a decomposition $\boldsymbol{S}^l = \boldsymbol{U}^l\boldsymbol{C}^l$ implied by Lemma 4.4. In this case the dynamics of the l states $\boldsymbol{\xi}^l$ taking values $\boldsymbol{\xi}^l(t) \in \mathbb{R}^{r^l}$ are

$$\dot{\boldsymbol{\xi}}^l(t) = \boldsymbol{C}^l\boldsymbol{v}\left(\boldsymbol{x}_0 + \sum_{l=1}^{L} \boldsymbol{U}^l\boldsymbol{\xi}^l(t)\right) \tag{4.13}$$

with initial conditions $\boldsymbol{\xi}^l(0) = \boldsymbol{0}$ and output map $\boldsymbol{x}^l = \boldsymbol{U}^l\boldsymbol{\xi}^l$. Differentiating $\boldsymbol{x}^l$ with respect to t recovers the layered dynamics (4.7). □

Therefore, although the layers' dynamics (4.7) have states $\boldsymbol{x}^l$ taking values in N-dimensional state-space, the layers are actually r^l-dimensional systems. Therefore, by choosing a decomposition $\boldsymbol{S} = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^L$ such that $\operatorname{rank}(\boldsymbol{S}^l) = r^l < r$ for all $l = 1,\ldots,L$, the resulting layers are each simpler subsystems in comparison to the full system. However, it is possible that certain (bad) choices of $\boldsymbol{S}^l$ can result in subsystems that are *more* complicated than the original high-dimensional system (1.2), whenever $r^l > r$. One way to ensure at least that $r^l \le r$ is to construct the layered decomposition by a partition of just one of the sets $\mathscr{S}$ or $\mathscr{R}$.

Definition 4.7 Suppose that the reaction set $\mathscr{R}$ has been partitioned into L disjoint subsets $\mathscr{R}_l$ for $l = 1,\ldots,L$. By defining the layered stoichiometric matrices $\boldsymbol{S}^l$ as

$$(\boldsymbol{S}^l)_{ij} = \begin{cases} (\boldsymbol{S})_{ij} & R_j \in \mathscr{R}_l, \\ 0 & \text{else,} \end{cases}$$

the layered decomposition is termed *layering by flux*. Similarly, by taking a partition of $\mathscr{S}$ into $\mathscr{S}_l$ for $l = 1, \dots, L$ the stoichiometric matrices

$$(\boldsymbol{S}^l)_{ij} = \begin{cases} (\boldsymbol{S})_{ij} & X_i \in \mathscr{S}_l, \\ 0 & \text{else}, \end{cases}$$

correspond to *layering by species*. In each of these cases, the fact that the subsets $\mathscr{R}_l$ and $\mathscr{S}_l$ respectively form a partition means that $\sum_{l=1}^{L} \boldsymbol{S}^l = \boldsymbol{S}$. $\diamondsuit$

When layering by flux, the column space inclusion $\mathrm{Col}(\boldsymbol{S}^l) \subseteq \mathrm{Col}(\boldsymbol{S})$ implies that $r^l \leq r$. Similarly, when layering by species, the row space inclusion $\mathrm{Row}(\boldsymbol{S}^l) \subseteq \mathrm{Row}(\boldsymbol{S})$ again implies that $r^l \leq r$. Another benefit of layering by flux or by species is that the resulting layers have clear interpretations in terms of the biochemically relevant concepts of species X_i and reactions R_j. Although other meaningful matrix sums $\boldsymbol{S} = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^L$ may exist, for now we consider 'layering' to mean 'layering by flux' or 'layering by species'. A schematic diagram of these two types of layered decomposition is depicted in the bottom two rows of Figure 4.1.

Consider again the partition of $\mathscr{R}$ in Example 4.1 resulting in the layered decomposition (4.10). The resulting stoichiometric matrices are such that $\mathrm{rank}(\boldsymbol{S}^1) = 2$ and $\mathrm{rank}(\boldsymbol{S}^2) = 3$. Both ranks are smaller than $\mathrm{rank}(\boldsymbol{S}) = 4$, and hence each subsystem is simpler than the original system.

4.3.2 Minimal layering

Layered decomposition, when carried out according to Definition 4.7, results in layers that are no more complicated than the original large-scale system. However, even though 'layering' has been characterised as layering by flux or by species, this begs the question of exactly *how* to partition the reaction set $\mathscr{R}$ or species set $\mathscr{S}$. One strategy would be to ensure that the resulting subsystems are as low-dimensional as possible, which we will term a *minimal layering*.

Proposition 4.8 (Rank sub-additivity [92, 124])**.** *For all matrices $\boldsymbol{M}_1, \dots, \boldsymbol{M}_K \in \mathbb{R}^{N \times M}$,*

$$\sum_{i=1}^{K} \mathrm{rank}(\boldsymbol{M}_i) \geq \mathrm{rank}\left(\sum_{i=1}^{K} \boldsymbol{M}_i \right). \tag{4.14}$$

Proposition 4.9 (Rank additivity [92, 124])**.** *Suppose $K = 2$. Then* (4.14) *holds with equality if and only if*

$$\dim\left[\mathrm{Col}(\boldsymbol{M}_1) \cap \mathrm{Col}(\boldsymbol{M}_2)\right] = \dim\left[\mathrm{Row}(\boldsymbol{M}_1) \cap \mathrm{Row}(\boldsymbol{M}_2)\right] = 0. \tag{4.15}$$

Corollary 4.10. *The total dimension $r^1 + \cdots + r^L \geq r$ of the layers defined by $\boldsymbol{S}^l$, for $l = 1, \dots, L$, is bounded below by the dimension of the original system. This inequality holds with equality, so that $r^1 + \cdots + r^L = r$, if and only if* (4.15) *holds for all possible choices of the two matrices $\boldsymbol{M}_1 = \sum_{l \in I_1} \boldsymbol{S}^l$ and $\boldsymbol{M}_2 = \sum_{l \in I_2} \boldsymbol{S}^l$ defined by disjoint sets $I_1, I_2 \subseteq \{1, \dots, L\}$ with $I_1 \cap I_2 = \emptyset$.*

Proof. The first statement of Corollary 4.10 follows directly from (4.14) with $\boldsymbol{M}_i = \boldsymbol{S}^i$ and $K = L$. To show the second statement, first assume that (4.15) holds for $\boldsymbol{M}_k = \sum_{l \in I_k} \boldsymbol{S}^l$ defined by all possible disjoint $I_1, I_2 \subseteq \{1,\dots,L\}$. It then follows that

$$\begin{aligned}
\operatorname{rank}(\boldsymbol{S}) &= \operatorname{rank}(\boldsymbol{S}^1 + (\boldsymbol{S}^2 + (\boldsymbol{S}^3 + (\cdots + (\boldsymbol{S}^L))))) \\
&= \operatorname{rank}(\boldsymbol{S}^1) + \operatorname{rank}(\boldsymbol{S}^2 + (\boldsymbol{S}^3 + (\cdots + (\boldsymbol{S}^L)))) \\
&\vdots \\
&= \operatorname{rank}(\boldsymbol{S}^1) + \operatorname{rank}(\boldsymbol{S}^2) + \cdots + \operatorname{rank}(\boldsymbol{S}^L).
\end{aligned}$$

To show the reverse implication, write $\boldsymbol{S} = \boldsymbol{M}_1 + \boldsymbol{M}_2 + \bar{\boldsymbol{S}}$, where each $\boldsymbol{M}_k = \sum_{l \in I_k} \boldsymbol{S}^l$, $k = 1,2$, is defined by arbitrary disjoint $I_1, I_2 \subseteq \{1,\dots,L\}$. Rank subadditivity (4.14) implies that

$$\begin{aligned}
\operatorname{rank}(\boldsymbol{S}) &\le \operatorname{rank}(\boldsymbol{M}_1 + \boldsymbol{M}_2) + \operatorname{rank}(\bar{\boldsymbol{S}}) \\
&\le \operatorname{rank}(\boldsymbol{M}_1) + \operatorname{rank}(\boldsymbol{M}_2) + \operatorname{rank}(\bar{\boldsymbol{S}}) \\
&\le \sum_{l \in \mathscr{I}_1} \operatorname{rank}(\boldsymbol{S}^l) + \sum_{l \in \mathscr{I}_2} \operatorname{rank}(\boldsymbol{S}^l) + \sum_{l \notin \mathscr{I}_1 \cup \mathscr{I}_2} \operatorname{rank}(\boldsymbol{S}^l) \\
&= \sum \operatorname{rank}(\boldsymbol{S}^l).
\end{aligned}$$

Hence assuming $\sum r^i = r$ implies that $\operatorname{rank}(\boldsymbol{M}_1 + \boldsymbol{M}_2) = \operatorname{rank}(\boldsymbol{M}_1) + \operatorname{rank}(\boldsymbol{M}_2)$, and the equivalence of the two conditions follows. □

When layering by flux, the row space condition in (4.15) is always satisfied. Similarly, when layering by species, the column space condition in (4.15) is always satisfied. Hence only one condition needs to be checked for each layering type.

Corollary 4.10 provides a condition on the column and row spaces of the stoichiometric matrices $\boldsymbol{S}^i$ that certifies whether the layering is minimal. In the case of layering by flux, this condition can be related to the space of steady-state distributions [153]. The following definition of the elementary flux modes (EFMs) [177] of a BRN makes use of the notation $\mathscr{R}(\boldsymbol{v}) = \{R_j \in \mathscr{R} \mid v_j > 0\}$ for the set of reactions R_j for which there is non-zero flux in the flux vector $\boldsymbol{v} = (v_1, \dots, v_M)^T$.

Definition 4.11 Consider a non-zero vector $\boldsymbol{v}$ satisfying the flux balance equation $\boldsymbol{S}\boldsymbol{v} = \boldsymbol{0}$ under the element-wise constraint $\boldsymbol{v} \ge \boldsymbol{0}$. If there do not exist vectors $\boldsymbol{v}' \ge \boldsymbol{0}$ and $\boldsymbol{v}'' \ge \boldsymbol{0}$ such that:

1. $\boldsymbol{S}\boldsymbol{v}' = \boldsymbol{0} = \boldsymbol{S}\boldsymbol{v}''$;

2. $\boldsymbol{v} = \alpha_1 \boldsymbol{v}' + \alpha_2 \boldsymbol{v}''$ for $\alpha_1, \alpha_2 > 0$;

3. the inclusions $\mathscr{R}(\boldsymbol{v}') \subseteq \mathscr{R}(\boldsymbol{v})$ and $\mathscr{R}(\boldsymbol{v}'') \subseteq \mathscr{R}(\boldsymbol{v})$ hold such that at least one inclusion is strict;

then $\boldsymbol{v}$ is an elementary flux mode (EFM).[1] ♢

Definition 4.11 uniquely determines the EFMs of a given BRN up to scalar multiplication [177]. In our formulation of BRN dynamics (1.2), we assumed that the flux vector $\boldsymbol{v} \ge \boldsymbol{0}$ is in the non-negative orthant by separating the forward and reverse fluxes of a reversible reaction into different 'reactions'. Trivially, an EFM can be defined by setting an equal flux $v_{j,f} = v_{j,r} > 0$ along both

[1] This definition was originally made in the context of metabolic networks, but can apply to any BRN with dynamics (1.2).

directions corresponding to each reversible reaction R_j, with all other fluxes $v_k = 0$, $k \neq j$. Thus, in any network with M_R reversible reactions, there are M_R EFMs corresponding to the forward and backward flux pairs.

The MATLAB toolbox `efmtool` can be used to compute all of the other EFMs of any given stoichiometric matrix [202]. If $\boldsymbol{v}$ is an EFM, we call $\mathscr{R}(\boldsymbol{v})$ a *minimal mass flow cycle*[2]. This may correspond to a cycle of mass flowing within a network, or the cycle of mass entering and leaving the system. The following result shows that a necessary condition for a minimal layering by flux is that no minimal mass flow cycle can be broken across layers.

Proposition 4.12. *Suppose that the partition of the reaction set into* $\{\mathscr{R}^l \mid l = 1,\dots,L\}$ *is a minimal layering by flux of a BRN. Then, for each EFM* $\boldsymbol{v}$ *of* $\boldsymbol{S}$*, the corresponding minimal mass flow cycle satisfies* $\mathscr{R}(\boldsymbol{v}) \subseteq \mathscr{R}^l$ *for exactly one value of* $l \in \{1,\dots,L\}$.

Proof. Consider an arbitrary EFM $\boldsymbol{v}$ with corresponding reactions $\mathscr{R}(\boldsymbol{v})$. By definition, $\boldsymbol{S}\boldsymbol{v} = \sum_{l=1}^{L} \boldsymbol{S}^l \boldsymbol{v} = \boldsymbol{0}$. For each $k = 1,\dots,L$, the relation $\boldsymbol{S}^k\boldsymbol{v} = -\sum_{l\neq k} \boldsymbol{S}^l \boldsymbol{v}$ implies that $\boldsymbol{S}^k \boldsymbol{v} \in \mathrm{Col}(\boldsymbol{S}^k) \cap \mathrm{Col}(\sum_{l\neq k} \boldsymbol{S}^l)$. Since the layering is minimal, it follows from Corollary 4.10 that $\boldsymbol{S}^k\boldsymbol{v} = \boldsymbol{0}$ for each $k = 1,\dots,L$.

Suppose, seeking contradiction, that there exists $K \in \mathbb{N}$ satisfying $2 \leq K \leq L$ such that (without loss of generality, for a given ordering of layer indices $l = 1,\dots,L$) the intersections $\mathscr{R}(\boldsymbol{v}) \cap \mathscr{R}^l \neq \emptyset$ for $l = 1,\dots,K$ and $\mathscr{R}(\boldsymbol{v}) \cap \mathscr{R}^l = \emptyset$ for $l > K$. This implies that the minimal mass flow cycle is broken across $K > 1$ layers. Define the two flux vectors

$$(\boldsymbol{v}')_j = \begin{cases} (\boldsymbol{v})_j & R_j \in \mathscr{R}(\boldsymbol{v}) \cap \mathscr{R}^1 \\ 0 & \text{else} \end{cases} \qquad (\boldsymbol{v}'')_j = \begin{cases} (\boldsymbol{v})_j & R_j \in \mathscr{R}(\boldsymbol{v}) \cap \left(\bigcup_{l=2}^{K} \mathscr{R}^l\right) \\ 0 & \text{else} \end{cases}$$

as the elements of the EFM $\boldsymbol{v}$ corresponding to the first layer and all the other layers respectively. By definition, $\boldsymbol{S}\boldsymbol{v}' = \boldsymbol{S}^1\boldsymbol{v}' = \boldsymbol{S}^1\boldsymbol{v} = \boldsymbol{0}$ and, similarly, $\boldsymbol{S}\boldsymbol{v}'' = \sum_{l=2}^{K} \boldsymbol{S}^l \boldsymbol{v}'' = \sum_{l=2}^{K} \boldsymbol{S}^l\boldsymbol{v} = \boldsymbol{0}$. Thus $\boldsymbol{v}' \geq \boldsymbol{0}$ and $\boldsymbol{v}'' \geq \boldsymbol{0}$ satisfy the first condition in Definition 4.11. Clearly, $\boldsymbol{v} = \boldsymbol{v}' + \boldsymbol{v}''$. Finally, we have constructed each flux vector such that both inclusions $\mathscr{R}(\boldsymbol{v}') = \mathscr{R}(\boldsymbol{v}) \cap \mathscr{R}^1 \subset \mathscr{R}(\boldsymbol{v})$ and $\mathscr{R}(\boldsymbol{v}'') = \mathscr{R}(\boldsymbol{v}) \cap \left(\bigcup_{l=2}^{K} \mathscr{R}^l\right) \subset \mathscr{R}(\boldsymbol{v})$ must hold strictly. Hence $\boldsymbol{v}$ is not an EFM, and we have a contradiction; it follows that $K = 1$. □

For example, consider again the layering of (4.9) carried out in Section 4.2. This layering is *not* minimal, since we can see from the ranks of $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ that $r^1 + r^2 = 5 > 4 = r$. The cycles of mass flow which correspond to the EFMs of this network can be found using `efmtool` [202] and are characterised by the four subsets of $\mathscr{R} = \{R_1,\dots,R_8\}$ given by $\{R_1, R_3, R_5, R_7, R_8\}$, $\{R_1, R_2\}$, $\{R_5, R_6\}$, and $\{R_1, R_3, R_4\}$. The only partition of $\mathscr{R}$ that respects the four minimal mass flow cycles is a trivial partition where $\mathscr{R}^1 = \mathscr{R}$. Hence there is no non-trivial minimal layering by flux of (4.9).

Recall the more restrictive definition of layering adopted in [32], where $\boldsymbol{S}$ is assumed to be block diagonal so that no mass is allowed to flow between layers. In our definition of a layered decomposition, a block-diagonal $\boldsymbol{S}$ is no longer necessary. However, in order to ensure that a layering by flux is a *minimal* layering, the block-diagonal condition is replaced by the steady-state condition that the reaction partition must respect the minimal mass flow cycles defined by the EFMs of the system.

[2] $\mathscr{R}(\boldsymbol{v})$ is also known as the *active set* of the EFM $\boldsymbol{v}$ [202].

4.3.3 Stability

Another important property of a layered decomposition (4.7) is its stability. A layering inherits a system's stability if, whenever the original system (1.2) taken from a particular initial condition $\boldsymbol{x}_0$ converges towards a steady state $\bar{\boldsymbol{x}}$, the state of each interconnected layer also converges towards its own steady state $\bar{\boldsymbol{x}}^l$. Note that in the following, we will abuse terminology slightly to say that if the layered states inherit the system's stability then the layering is *stable*; however, we make no claims about the stability properties of each layer's equilibrium. The following proposition shows that this property is a consequence of a minimal layering.

Proposition 4.13. *Suppose the ODE system* (1.2) *is stable, with limiting value* $\boldsymbol{x}(t) \to \bar{\boldsymbol{x}}$ *as* $t \to \infty$*. Suppose also that the stoichiometric matrices* $\boldsymbol{S} = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^L$ *define a minimal layering* (4.7)*. Then there exist L vectors* $\bar{\boldsymbol{x}}^l$ *such that* $\boldsymbol{x}^l(t) \to \bar{\boldsymbol{x}}^l$ *as* $t \to \infty$*, which are unique up to the distribution of initial conditions* $\boldsymbol{x}_0^l$ *amongst the layers.*

Proof. Recall the minimal realisation (4.13) of the layered dynamics, with states $\boldsymbol{\xi}^l$. Define the state $\boldsymbol{\xi}^T = [\boldsymbol{\xi}^1, \dots, \boldsymbol{\xi}^L]$, taking values $\boldsymbol{\xi}(t) \in \mathbb{R}^{\sum_l r^l} = \mathbb{R}^r$. The dynamics of this combined state are

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{C}\boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{U}\boldsymbol{\xi}(t)\big),$$

with initial conditions $\boldsymbol{\xi}(0) = \boldsymbol{0}$, where

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{C} = \begin{bmatrix} \boldsymbol{U}^1 & \dots & \boldsymbol{U}^L \end{bmatrix} \begin{bmatrix} \boldsymbol{C}^1 \\ \vdots \\ \boldsymbol{C}^L \end{bmatrix}.$$

This matches the reduced dynamics (4.12), meaning that the output map $\boldsymbol{x}(t) = \boldsymbol{x}_0 + \boldsymbol{U}\boldsymbol{\xi}(t)$ recovers the trajectory of $\boldsymbol{x}$ given by the original dynamics (1.2).

The columns of $\boldsymbol{U}$ are linearly independent since it is formed from $\boldsymbol{U}^l$ corresponding to a minimal layering, and hence the output map $\boldsymbol{x}(t) = \boldsymbol{x}_0 + \boldsymbol{U}\boldsymbol{\xi}(t)$ is invertible. Since $\boldsymbol{x}(t) \to \bar{\boldsymbol{x}}$ as $t \to \infty$, this implies that there exists a unique vector $\bar{\boldsymbol{\xi}} \in \mathbb{R}^r$ such that $\boldsymbol{\xi}(t) \to \bar{\boldsymbol{\xi}}$ as $t \to \infty$. Decomposing $\bar{\boldsymbol{\xi}}^T = [\bar{\boldsymbol{\xi}}^1, \dots, \bar{\boldsymbol{\xi}}^L]$ according to the construction of $\boldsymbol{\xi}$ above defines for each layer $l = 1, \dots, L$ a corresponding vector $\bar{\boldsymbol{\xi}}^l \in \mathbb{R}^{r^l}$. Clearly, $\boldsymbol{\xi}(t) \to \bar{\boldsymbol{\xi}}$ implies that each $\boldsymbol{\xi}^l(t) \to \bar{\boldsymbol{\xi}}^l$. Applying the output map from $\boldsymbol{\xi}^l$ to $\boldsymbol{x}^l$ gives

$$\boldsymbol{x}^l(t) = \boldsymbol{x}_0^l + \boldsymbol{U}^l\boldsymbol{\xi}^l(t) \to \boldsymbol{x}_0^l + \boldsymbol{U}^l\bar{\boldsymbol{\xi}}^l = \bar{\boldsymbol{x}}^l,$$

for a uniquely defined $\bar{\boldsymbol{x}}^l$, given each $\boldsymbol{x}_0^l$. Hence each layer is guaranteed to approach the steady state $\bar{\boldsymbol{x}}^l$ when the overall system is stable. □

A minimal layering is sufficient for a stable layering (i.e. the guarantee that each layer approaches steady state if the system is stable) but it is not necessary. For example, suppose that the system is layered by species. Then $x_i(t) \to \bar{x}_i$ for each $i = 1, \dots, N$ implies that $\boldsymbol{x}^l \to \bar{\boldsymbol{x}}^l$, since in a layering by species only one of the contributions $x_i^l(t)$, $l = 1, \dots, L$ is non-zero. This property is independent of whether the layering is minimal. However, if a different type of layering (i.e. a layering by flux) is not minimal, then certain components x_i^l may approach infinity, but be cancelled out by a similar unboundedness towards negative infinity in other components x_i^j, $j \neq l$.

This can be seen in Figure 4.2, where $x^1 \to \infty$ and $x^2 \to -\infty$, but $x = x^1 + x^2$ approaches a steady state.

The following section discusses an adaptation to the technique of layering by flux that always defines minimal, stable layerings. Similarly to layering by flux (according to Definition 4.7), the minimal layers are defined from a partition of the reaction set $\mathscr{R}$.

4.4 Flux Layer Minimisation

Consider the dynamics (1.2) of a BRN, and assume that the state $\boldsymbol{x}(t)$ converges to $\bar{\boldsymbol{x}}$ as $t \to \infty$. Assume also that we are given a partition of the reaction set $\mathscr{R}$ into L disjoint sets. In this section, we propose an algorithm that uses the partition of $\mathscr{R}$ to produce a layering that is guaranteed to be minimal. The result of this algorithm will coincide with a standard layering by flux, according to Definition 4.7, if the latter produces a minimal layering.

4.4.1 Two layers

We first consider the case where $L = 2$. For the stoichiometric matrices $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ defined by the standard layering by flux technique introduced above, consider the linear space $\mathrm{Col}(\boldsymbol{S}^1) \cap \mathrm{Col}(\boldsymbol{S}^2)$ given by the intersection of their two column spaces. If this space is of dimension 0 then layering by flux produces a minimal layering. However, suppose that this is a non-minimal layering, meaning that the column-space intersection is a non-trivial space. In this case, the vectors in this space correspond to the directions in which each of the translated states $\boldsymbol{x}^l(t)$ may take unbounded values.

The following result will exploit the decomposition $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$ of the layered stoichiometric matrices. Let $\boldsymbol{U}_{1,2} \in \mathbb{R}^{N \times r_{1,2}}$ be a matrix of $r_{1,2} = r^1 + r^2 - r$ columns that form a basis for the intersection space $\mathrm{Col}(\boldsymbol{S}^1) \cap \mathrm{Col}(\boldsymbol{S}^2)$. This basis can be extended for each $l = 1, 2$ into a basis of size r^l for the column space $\mathrm{Col}(\boldsymbol{S}^l)$. These bases define the columns of the matrices

$$\boldsymbol{U}^l = \begin{bmatrix} \hat{\boldsymbol{U}}^l & \boldsymbol{U}_{1,2} \end{bmatrix},$$

for $l = 1, 2$. The final $r_{1,2}$ columns of each $\boldsymbol{U}^l$ are the same, corresponding to the basis of $\mathrm{Col}(\boldsymbol{S}^1) \cap \mathrm{Col}(\boldsymbol{S}^2)$. Given each $\boldsymbol{U}^l$, a corresponding $\boldsymbol{C}^l$ exists such that $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$. This can be written conformally with the decomposition of $\boldsymbol{U}^l$ above as

$$\boldsymbol{C}^l = \begin{bmatrix} \hat{\boldsymbol{C}}^l \\ \boldsymbol{C}^l_{1,2} \end{bmatrix}.$$

Unlike each $\boldsymbol{U}^l$, the final $r_{1,2}$ rows of $\boldsymbol{C}^1$ and $\boldsymbol{C}^2$ are not equal. Based on these expressions for $\boldsymbol{U}^l$ and $\boldsymbol{C}^l$, define the layered dynamics

$$\dot{\boldsymbol{z}}^1(t) = \left[\hat{\boldsymbol{U}}^1 \hat{\boldsymbol{C}}^1 + \boldsymbol{U}_{1,2} \boldsymbol{C}_{1,2}\right] \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{z}^1(t) + \boldsymbol{z}^2(t)\big), \tag{4.16a}$$

$$\dot{\boldsymbol{z}}^2(t) = \left[\hat{\boldsymbol{U}}^2 \hat{\boldsymbol{C}}^2\right] \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{z}^1(t) + \boldsymbol{z}^2(t)\big), \tag{4.16b}$$

where $\boldsymbol{C}_{1,2} = \boldsymbol{C}^1_{1,2} + \boldsymbol{C}^2_{1,2}$. The following result proves that this new layering, based on the original layering by flux, is minimal and therefore stable.

Lemma 4.14. *The new stoichiometric matrices* $\hat{\boldsymbol{S}}^1 = \hat{\boldsymbol{U}}^1\hat{\boldsymbol{C}}^1 + \boldsymbol{U}_{1,2}\boldsymbol{C}_{1,2}$ *and* $\hat{\boldsymbol{S}}^2 = \hat{\boldsymbol{U}}^2\hat{\boldsymbol{C}}^2$ *used for the layered dynamics* (4.16) *define a minimal layering, with* $\operatorname{rank}(\hat{\boldsymbol{S}}^1) = r^1$ *and* $\operatorname{rank}(\hat{\boldsymbol{S}}^2) = r^2 - r_{1,2} = r - r^1$.

Proof. By the definition of the stoichiometric matrices,

$$\begin{aligned}\hat{\boldsymbol{S}}^1 + \hat{\boldsymbol{S}}^2 &= \hat{\boldsymbol{U}}^1\hat{\boldsymbol{C}}^1 + \boldsymbol{U}_{1,2}\boldsymbol{C}_{1,2} + \hat{\boldsymbol{U}}^2\hat{\boldsymbol{C}}^2 \\ &= \big(\hat{\boldsymbol{U}}^1\hat{\boldsymbol{C}}^1 + \boldsymbol{U}_{1,2}\boldsymbol{C}^1_{1,2}\big) + \big(\hat{\boldsymbol{U}}^2\hat{\boldsymbol{C}}^2 + \boldsymbol{U}_{1,2}\boldsymbol{C}^2_{1,2}\big) \\ &= \boldsymbol{S}^1 + \boldsymbol{S}^2 = \boldsymbol{S},\end{aligned}$$

and hence $\hat{\boldsymbol{S}}^1$ and $\hat{\boldsymbol{S}}^2$ form a layered decomposition of (1.2). The columns and rows of $\hat{\boldsymbol{U}}^2 \in \mathbb{R}^{N\times(r^2-r_{1,2})}$ and $\hat{\boldsymbol{C}}^2 \in \mathbb{R}^{(r^2-r_{1,2})\times M}$ respectively are linearly independent, and therefore form bases for the column space and row space of $\hat{\boldsymbol{S}}^2 = \hat{\boldsymbol{U}}^2\hat{\boldsymbol{C}}^2$. Hence $\operatorname{rank}(\hat{\boldsymbol{S}}^2) = r^2 - r_{1,2} = r - r^1$.

The other layer's stoichiometry is given by $\hat{\boldsymbol{S}}^1 = \boldsymbol{U}^1\big(\boldsymbol{C}^1 + [\boldsymbol{0}, \boldsymbol{C}^2_{1,2}]^T\big)$, where the matrix $\boldsymbol{U}^1 = [\hat{\boldsymbol{U}}^1, \boldsymbol{U}_{12}] \in \mathbb{R}^{N\times r^1}$ has rank r^1. Therefore $\operatorname{rank}(\hat{\boldsymbol{S}}^1) \le \operatorname{rank}(\boldsymbol{U}^1) \le r^1$. However, rank subadditivity (4.14) implies that $\operatorname{rank}(\hat{\boldsymbol{S}}^1) \ge \operatorname{rank}(\boldsymbol{S}) - \operatorname{rank}(\hat{\boldsymbol{S}}^2) = r^1$, and it follows that $\operatorname{rank}(\hat{\boldsymbol{S}}^1) = r^1$.

Since $\operatorname{rank}(\hat{\boldsymbol{S}}^1) + \operatorname{rank}(\hat{\boldsymbol{S}}^2) = \operatorname{rank}(\boldsymbol{S})$, the modified dynamics (4.16) define a minimal layering. □

To illustrate this minimal layering, consider the reaction partition used in Example 4.1 to construct the non-minimal layering by flux of the gene-transcription cascade (4.1). The stoichiometric matrices (4.10) have intersecting column spaces given by $\mathrm{Col}(\boldsymbol{S}^1)\cap\mathrm{Col}(\boldsymbol{S}^2) = \mathrm{span}([0,1,0,0]^T)$. Applying (4.16) to produce guaranteed minimal stoichiometric matrices from the given reaction partition gives

$$\hat{\boldsymbol{S}}^1 = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \hat{\boldsymbol{S}}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

Each matrix $\hat{\boldsymbol{S}}^l$ clearly has rank 2. Therefore this decomposition is a minimal layering of (4.9). In fact, this layered decomposition corresponds to a layering by species, rather than by flux. It is important to note, however, that the minimal layering (4.16) produced by any partition of $\mathscr{R}$ is, in general, neither a layering by flux nor by species.

4.4.2 More layers

Now suppose that there is a given partition of the reaction set $\mathscr{R}$ into an ordered collection of $L > 2$ disjoint sets $\mathscr{R}_l$, for $l = 1,\dots,L$. The following *Flux Layer Minimisation algorithm* produces L stoichiometric matrices $\hat{\boldsymbol{S}}^l$, $l = 1,\dots,L$, that define a minimal layering.

Algorithm 4.15 (Flux Layer Minimisation)

```
1     Set $\boldsymbol{U}^L_{AB} = \boldsymbol{0} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{C}^L_{AB} = \boldsymbol{0} \in \mathbb{R}^{1\times M}$
2     Define $\boldsymbol{S}^l$ according to layering by flux (Definition 4.7) for $l = 1,\dots,L$.
3     For $l = L : -1 : 2$
3.1       Set $\boldsymbol{A}^l = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^{l-1}$ and $\boldsymbol{B}^l = \boldsymbol{S}^l + \boldsymbol{U}^l_{AB}\boldsymbol{C}^l_{AB}$
3.2       Define the columns of $\boldsymbol{U}^l_{AB}$ as a basis for $\mathrm{Col}(\boldsymbol{A}^l) \cap \mathrm{Col}(\boldsymbol{B}^l)$
3.3       Extend to $\boldsymbol{U}^l = [\hat{\boldsymbol{U}}^l, \boldsymbol{U}^l_{AB}]$, as the basis of $\mathrm{Col}(\boldsymbol{B}^l)$
3.4       Solve $\boldsymbol{B}^l = \boldsymbol{U}^l\boldsymbol{C}^l$ to find $\boldsymbol{C}^l = [\hat{\boldsymbol{C}}^l, \boldsymbol{C}^l_{AB}]^T$
3.5       Set $\hat{\boldsymbol{S}}^l = \hat{\boldsymbol{U}}^l\hat{\boldsymbol{C}}^l$
3.6       Loop
4     Set $\hat{\boldsymbol{S}}^1 = \boldsymbol{S}^1 + \boldsymbol{U}^2_{AB}\boldsymbol{C}^2_{AB}$
5     End
```

The difference between the stoichiometric matrices $\hat{\boldsymbol{S}}^l$ produced by this algorithm and those produced by (non-minimal) layering by flux can be interpreted as defining a set of *correction fluxes* between layers. The layered decomposition given by each $\hat{\boldsymbol{S}}^l$ can be expressed in terms of the non-minimal stoichiometric matrices by writing

$$\begin{aligned}\dot{\boldsymbol{z}}^l &= \hat{\boldsymbol{S}}^l\boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{z}^1 + \cdots + \boldsymbol{z}^L)\\ &= \left[\left(\boldsymbol{S}^l + \boldsymbol{U}^{l+1}_{AB}\boldsymbol{C}^{l+1}_{AB}\right) - \boldsymbol{U}^l_{AB}\boldsymbol{C}^l_{AB}\right]\boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{z}^1 + \cdots + \boldsymbol{z}^L),\end{aligned}$$

for $l = 1,\dots,L$, where we define the matrices $\boldsymbol{U}^1_{AB}\boldsymbol{C}^1_{AB} = \boldsymbol{U}^{L+1}_{AB}\boldsymbol{C}^{L+1}_{AB} = \boldsymbol{0}$. Thus each $\boldsymbol{U}^l_{AB}\boldsymbol{C}^l_{AB}\boldsymbol{v}$ defines a flux from layer l to layer $l-1$ that stabilises layer l, given that layer l is already receiving the correction flux $\boldsymbol{U}^{l+1}_{AB}\boldsymbol{C}^{l+1}_{AB}\boldsymbol{v}$ from layer $l+1$. The effects of the inter-layer correction fluxes are not 'seen' in the system trajectory $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{l=1}^{L}\boldsymbol{x}^l$, since they cancel one another out when aggregated.

Proposition 4.16. *The layered dynamics defined by the stoichiometric matrices resulting from the Flux Layer Minimisation algorithm given by*

$$\dot{\boldsymbol{z}}^l = \hat{\boldsymbol{S}}^l\boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{z}^1 + \cdots + \boldsymbol{z}^L), \tag{4.17}$$

define a minimal layering.

Proof. This result can be proven inductively by Lemma 4.14. At the first stage, where $l = L$, the matrices $\boldsymbol{A}^L$ and $\boldsymbol{B}^L$ define a non-minimal layering of (1.2) into two layers. By Lemma 4.14, the two matrices $\hat{\boldsymbol{S}}^L$ and $\left(\boldsymbol{A}^{L-1} + \boldsymbol{B}^{L-1}\right)$ given by the first stage of the Flux Layer Minimisation algorithm define a minimal layering of the system into two layers. If $\mathrm{rank}(\hat{\boldsymbol{S}}^L) = r^L$, then the dimension of the other layer is $r - \hat{r}^L$.

Consider the BRN with stoichiometric matrix $\left(\boldsymbol{A}^{L-1} + \boldsymbol{B}^{L-1}\right)$. The matrices $\boldsymbol{A}^{L-1}$ and $\boldsymbol{B}^{L-1}$ define a non-minimal layering of this system. As in the previous step, Lemma 4.14 implies that the matrices $\hat{\boldsymbol{S}}^{L-1}$ and $\left(\boldsymbol{A}^{L-2} + \boldsymbol{B}^{L-2}\right)$ resulting from the second stage of the algorithm, where $l = L-1$, define a minimal layering of this subsystem into two layers. If $\mathrm{rank}(\hat{\boldsymbol{S}}^{L-1}) = r^{L-1}$ then the dimension of the other layer is $(r - \hat{r}^L) - r^{L-1}$.

Continuing this procedure iteratively to $L = 2$, we find that the dimension of $\hat{\boldsymbol{S}}^1$ is equal to $(((r - \hat{r}^L) - \hat{r}^{L-1}) - \dots) - \hat{r}^2$. Hence $r = \sum_{l=1}^{L}\hat{r}^l$, where $\hat{r}^l = \mathrm{rank}(\hat{\boldsymbol{S}}^l)$ for $l = 1,\dots,L$, and therefore the stoichiometric matrices $\hat{\boldsymbol{S}}^l$ define a minimal layering. □

Proposition 4.16 proves that the Flux Layer Minimisation algorithm derives a minimal, and therefore stable, layering from any partition of the reaction set $\mathscr{R}$. If the standard procedure for layering by flux given in Definition 4.7 gives a minimal layering, then the Flux Layer Minimisation algorithm does not modify this decomposition.

4.5 Examples

4.5.1 Enzyme dynamics

Recall the BRN (3.26) with dynamics (3.27) described in Section 3.4.3. It was suggested that a decomposition of this model would help make it more amenable to analysis. Suppose this system is decomposed through layering by flux into two layers. Layer 1 consists of the reactions in (3.26a)–(3.26b), corresponding to the first 3 columns of $\boldsymbol{S}$, while layer 2 is made up of the reactions in (3.26c)–(3.26d), corresponding to the other 4 columns, giving

$$\boldsymbol{S}^1+\boldsymbol{S}^2=\begin{bmatrix} -1&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0\\ 1&0&-1&0&0&0&0\\ 0&1&-1&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{bmatrix}+\begin{bmatrix} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&-1&1&1&0\\ 0&0&0&-1&1&0&0\\ 0&0&0&1&-1&-1&0\\ 0&0&0&0&0&1&-1\\ 0&0&0&0&0&0&-1\\ 0&0&0&0&0&0&1 \end{bmatrix}. \tag{4.18}$$

This layering is minimal, in the sense that $\operatorname{rank}(\boldsymbol{S})=6=3+3=\operatorname{rank}(\boldsymbol{S}^1)+\operatorname{rank}(\boldsymbol{S}^2)$.

Thus, if the original system is stable then the layers are also stable. From the initial conditions $\boldsymbol{x}_0$ given in Section 3.4.3, the system (3.26) is stable such that $\boldsymbol{x}\to\bar{\boldsymbol{x}}$ given by

$$\bar{\boldsymbol{x}}=\begin{bmatrix}0 & 0 & (A_1-P) & (A_2-P) & P & 0 & 0 & (X_0+Y_0-R) & (K_0-R) & R\end{bmatrix}^T.$$

Hence, given the initial conditions $\boldsymbol{x}_0^l=\boldsymbol{0}$ for $l=1,2$ and the layered dynamics (4.7), there are unique vectors $\bar{\boldsymbol{x}}^l$ such that $\boldsymbol{x}^l(t)\to\bar{\boldsymbol{x}}^l$ as $t\to\infty$, for $l=1,2$, with $\boldsymbol{x}_0+\bar{\boldsymbol{x}}^1+\bar{\boldsymbol{x}}^2=\bar{\boldsymbol{x}}$.

The zero rows of $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ imply that $x^1=c^1=y^1=k^1=z^1=0$ in layer 1, and $p_1^2=p_2^2=e_1^2=e_2^2=0$ in layer 2. Neglecting the zero rows and columns of each $\boldsymbol{S}^l$, the first layer's dynamics are

$$\begin{bmatrix}\dot{p}_1^1\\ \dot{p}_2^1\\ \dot{e}_1^1\\ \dot{e}_2^1\\ \dot{e}^1\end{bmatrix}=\begin{bmatrix}-1&0&0\\ 0&-1&0\\ 1&0&-1\\ 0&1&-1\\ 0&0&1\end{bmatrix}\begin{bmatrix}\alpha_1\left(A_1+p_1^1\right)\\ \alpha_2\left(A_2+p_2^1\right)\\ \alpha_{12}e_1^1e_2^1\end{bmatrix} \tag{4.19a}$$

with zero initial conditions and steady state $\left(-A_1,-A_2,(A_1-P),(A_2-P),P\right)^T$, for constant $P=$

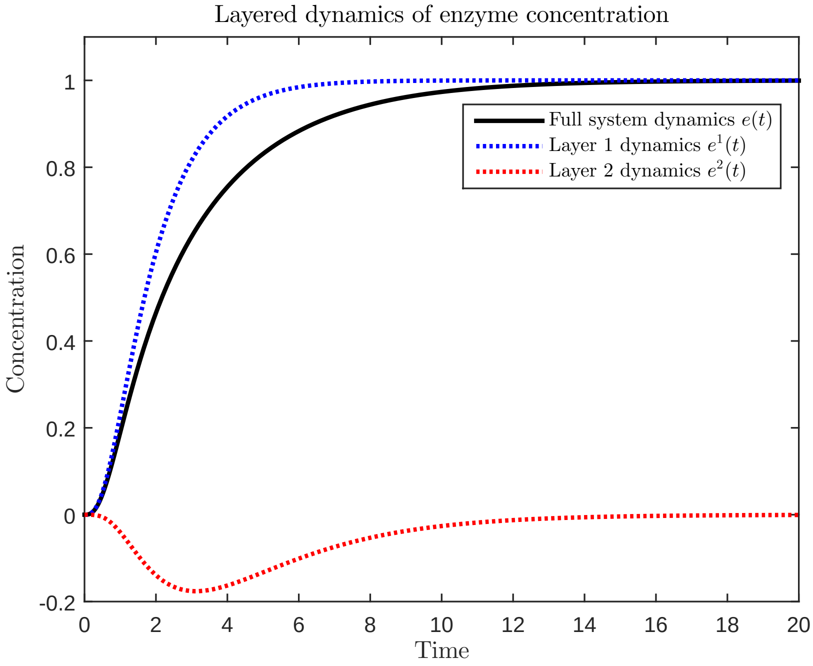


Figure 4.3: Three simulations of the trajectory $e(t)$, for parameters $\alpha_i = \beta_i = 1$ and initial conditions $\boldsymbol{x}(0) = [2,1,0,0,0,1,0,0,2,0]^T$. The black solid line is the trajectory of $e(t)$ given by (3.27); the broken lines are the layered contributions $e^1(t)$ and $e^2(t)$ from (4.19) that each sum to give $e(t)$.

$\min(A_1, A_2)$. The second layer has dynamics

$$\begin{bmatrix} \dot{e}^2 \\ \dot{x}^2 \\ \dot{c}^2 \\ \dot{y}^2 \\ \dot{k}^2 \\ \dot{z}^2 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \left(X_0 + x^2\right)\left(e^1 + e^2\right) \\ \beta_{-1} c^2 \\ \beta_2 c^2 \\ \beta_3 \left(K_0 + k^2\right)\left(Y_0 + y^2\right) \end{bmatrix} \tag{4.19b}$$

with zero initial conditions and steady state $\left(0, -X_0, 0, (X_0 - R), -R, R\right)^T$ for $R = \min(X_0 + Y_0, K_0)$.

The dynamics of e have non-zero contributions e^1 and e^2 from both layers. These contributions are summed to give the trajectory of e in the integrated system, as shown in Figure 4.3. However, unlike Example 4.1, this system is a cascade, since (4.19a) is independent of all components of $\boldsymbol{x}^2$. Hence, the simulation and analysis of this system is decomposed such that the three-dimensional upstream layer can be considered in isolation first. The resulting trajectory of e^1 forms the input to the downstream dynamics.

Suppose that the modular framework is used to decompose (3.26) into modules, instead of layers. This requires a block-diagonal decomposition of $\boldsymbol{S}$ in (3.27), corresponding to a simultaneous partition of the species set $\mathscr{S}$ and reaction set $\mathscr{R}$. For the same partition of $\mathscr{R}$ as in the layered decomposition, suppose that $\mathscr{S}^1 = \{P_1, P_2, E_1, E_2, E\}$ and $\mathscr{S}^2 = \{X, C, Y, K, Z\}$. This corresponds to putting the first 5 × 3 block of $\boldsymbol{S}$ as the upstream module's stoichiometry, and the next 5 × 4 block as the downstream module's stoichiometry, as in the top row of Figure 4.1. The concentration e of E is the input from the upstream to the downstream module. There are non-zero elements of $\boldsymbol{S}$ that are neglected by this decomposition, which implies that a retroactive interconnection effect will result.

The upstream module follows the same dynamics (4.19a) as the upstream layer. Hence the trajectory of e in the isolated module is given by the broken blue curve in Figure 4.3. As expected, however, there is also a retroactive effect, since the trajectory of e given by the integrated dynamics (3.27) is not equal to the isolated module's trajectory (as illustrated by the solid black curve in Figure 4.3). In fact, the effect of the retroactivity is exactly equal to the trajectory e^2 of the downstream layer.

The layered decomposition (4.19) has a cascade structure, whereas retroactivity in the modular decomposition implies that the modules are in feedback. Thus this example illustrates how layered decomposition can uncover cascade structures that modular decomposition cannot. Chapter 6 will discuss layered interconnection structures in greater detail.

4.5.2 Glycolysis

As a second example of layered decomposition, consider the glycolysis model constructed by Hynne *et al* [94]. This system consists of 22 biochemical species $X_i \in \mathscr{S}$ taking part in 24 reactions $R_j \in \mathscr{R}$. The model has been implemented by the authors in SBML [93] and can be downloaded from the Biomodels database [117]. A number of the reactions are reversible; as discussed in Section 4.3.2, there are trivial EFMs $\boldsymbol{u}_j$ with corresponding minimal mass flow cycles $\mathscr{R}(\boldsymbol{u}_j) = \{R_{j,f}, R_{j,r}\}$ corresponding to the forward and reverse directions respectively of the reversible reaction R_j.

An EFM analysis using `efmtool` [202] finds a further four overlapping EFMs $\boldsymbol{v}_i^\star$, $i = 1,2,3,4$, with the corresponding reactions

$$\begin{aligned}
\mathscr{R}(\boldsymbol{v}_1^\star) &= \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{23}\},\\
\mathscr{R}(\boldsymbol{v}_2^\star) &= \{R_1, R_2, R_3, R_4, R_5, R_6, R_8, R_9, R_{10}, R_{11}, R_{15}, R_{16}, R_{17}, R_{18}, R_{20}, R_{21}\},\\
\mathscr{R}(\boldsymbol{v}_3^\star) &= \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{22}\},\\
\mathscr{R}(\boldsymbol{v}_4^\star) &= \{R_1, R_2, R_3, R_4, R_5, R_6, R_8, R_9, R_{10}, R_{11}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}\}.
\end{aligned}$$

The sets above slightly abuse notation: by writing $R_j \in \mathscr{R}(\boldsymbol{v}_k^\star)$ for a reversible reaction, we mean that exactly one of the forward or reverse reactions $R_{j,f}, R_{j,r} \in \mathscr{R}(\boldsymbol{v}_k^\star)$ respectively. Any partition of $\mathscr{R}$ that gives a minimal (and hence stable) layering by flux must respect each of the sets $\mathscr{R}(\boldsymbol{v}_i^\star)$ for $i = 1,2,3,4$, plus the trivial pairs $\mathscr{R}(\boldsymbol{u}_j)$ for reversible R_j. This implies that all reactions apart from R_{24} must be in a single layer, which is not a particularly useful decomposition.

However, the Flux Layer Minimisation algorithm described in Section 4.4 can be used to derive a minimal, stable layering from any partition of the reaction set $\mathscr{R}$. For example, consider the partition $\mathscr{R}^1 = \{R_{23}, R_{24}\}$, $\mathscr{R}^2 = \{R_j \mid j = 15, \ldots, 21\}$, and $\mathscr{R}^3 = \mathscr{R}(\boldsymbol{v}_3^\star)$. The non-zero pattern of the stoichiometric matrices $\boldsymbol{S}^l$ resulting from layering by flux (according to Definition 4.7) are shown in the left-hand column of Figure 4.4. The ranks $r^l = \text{rank}(\boldsymbol{S}^l)$ of these stoichiometric matrices are respectively $(r^1, r^2, r^3) = (2, 6, 14)$. The difference $\sum_{l=1}^{3} r^l - r = 22 - 20 = 2$ shows that this is a non-minimal layering. If we instead derive the layered stoichiometric matrices through the Flux Layer Minimisation algorithm, the resulting stoichiometric matrices $\hat{\boldsymbol{S}}^l$ have ranks $(\hat{r}^1, \hat{r}^2, \hat{r}^3) = (2, 6, 12)$. The ranks $\hat{r}^l$ sum to 20, and therefore $\hat{\boldsymbol{S}}^l$, $l = 1, 2, 3$, define a minimal layering of the system. The non-zero patterns of these matrices are shown in the right-hand column of Figure 4.4.

Layer 1, given by $\hat{\boldsymbol{S}}^1$, is identified with species X_3, X_5, and X_{25}, representing the co-factors ATP, ADP, and AMP. However, the minimal layering is much less easily interpretable in biochemical terms than the original definition of layering by flux, since there is no partition of either $\mathscr{R}$ or $\mathscr{S}$ amongst the subsystems. There are additional non-zero stoichiometric coefficients $(\hat{\boldsymbol{S}}^l)_{ij} \neq 0$ in $\hat{\boldsymbol{S}}^2$ and $\hat{\boldsymbol{S}}^3$ that make certain species X_i dependent on fluxes v_j, even though they do not take any part in reaction R_j in the original network. However, as discussed in Section 4.4.2, these stoichiometric coefficients can be interpreted as a model of mass flow between corresponding unstable x_i^2 and x_i^3, ensuring that the layered states $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$ are stable in the minimal layering.

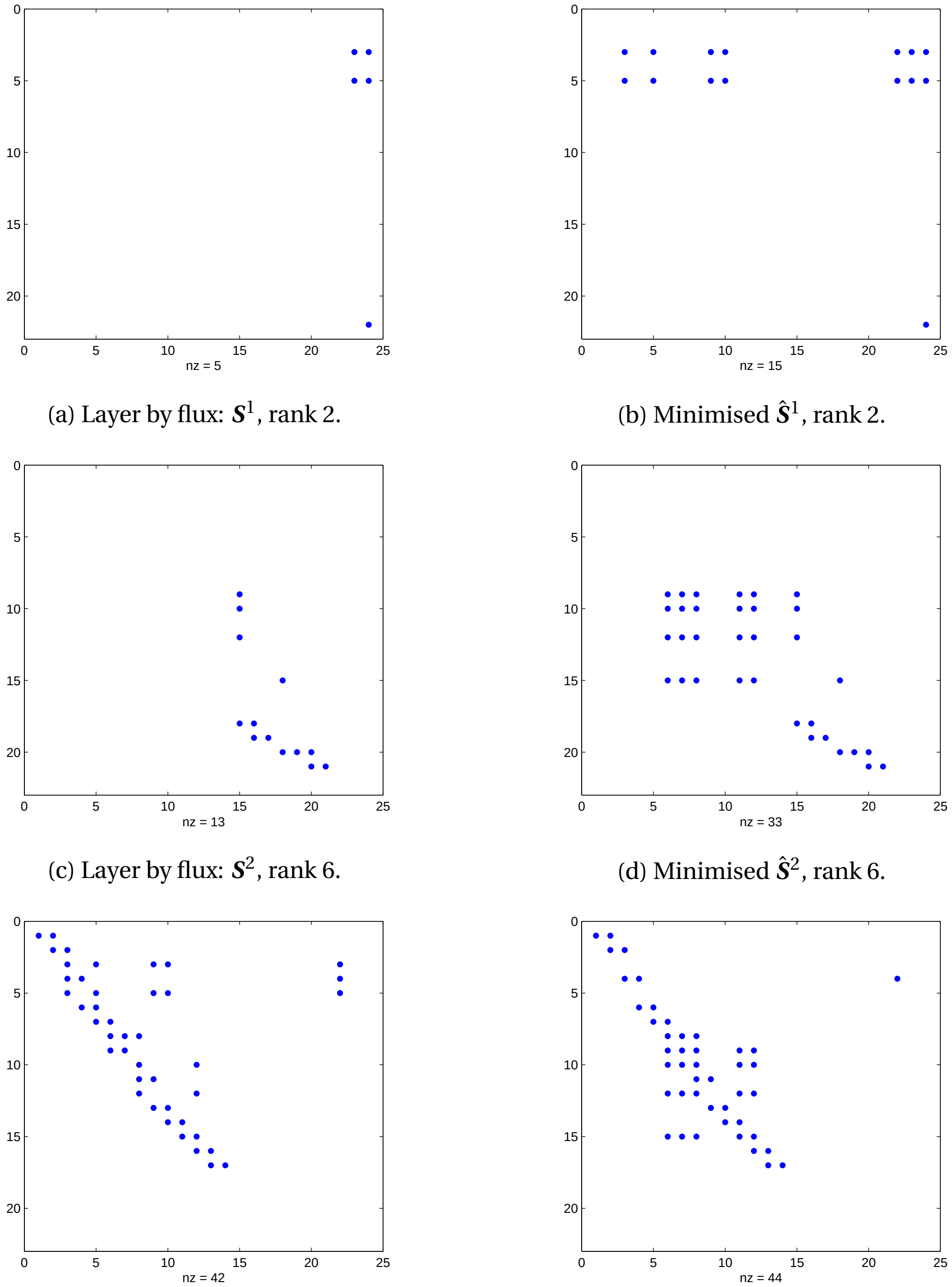
4.6 Discussion

This chapter has introduced the layering framework as a novel method of decomposing large-scale BRN models (1.2). Rather than partitioning the state space into components, as with modular approaches, the layered approach instead replicates the state vector into each of the subsystems. Unlike modules, the input–output behaviour (i.e. the dynamics) of any layer will not change upon interconnection with other layers, although we are careful to distinguish between the dynamics and the trajectory of each layer. Consider the examples in Example 4.1 and Section 4.5.1, that compare the modular and layered frameworks. In each case, the dynamics of layer 1 are equal to the isolated dynamics of module 1: however, when module 2 is integrated with the system, module 1 needs to be remodelled. This is not the case in the layered framework, where the input–output dynamics of layer 1 are not affected by the presence of layer 2; the presence of layer 2 is modelled by a change in the input to layer 1.

The dimension of a BRN is identified as the rank of its stoichiometric matrix $\boldsymbol{S}$. This property can be used to test whether a particular decomposition results in lower-dimensional layers. Two types of layered decomposition were defined: layering by species, and layering by flux. In these cases, the dimensions of the resulting subsystems are guaranteed to be no larger than that of the original system. An important question we can ask of a given layered decomposition is whether it inherits the full system's stability, so that the layered states approach steady state whenever the original state does. This is always the case if layering by species. It is also always the case if the layering is minimal, meaning that the layers' dimensions are as small as possible.

Often when layering by flux the resulting decomposition is not minimal. In this case, the Flux Layer Minimisation algorithm, introduced in Section 4.4, can be used to produce stoichiometric matrices that define a minimal layering from any given partition of the reaction set. However, as shown in Section 4.5.2 this decomposition is not necessarily particularly intuitive, especially when compared to the stoichiometric matrices resulting from layering by flux or species. If, instead, the decomposition strategy is constrained to only consider a minimal layering by flux, then the number of possible decompositions is drastically reduced; indeed, the EFMs may imply that such a decomposition is infeasible.

The relaxation of the block-diagonal condition of [32] to the EFM condition given in this



(a) Layer by flux: $\boldsymbol{S}^1$, rank 2.

(b) Minimised $\hat{\boldsymbol{S}}^1$, rank 2.

(c) Layer by flux: $\boldsymbol{S}^2$, rank 6.

(d) Minimised $\hat{\boldsymbol{S}}^2$, rank 6.

(e) Layer by flux: $\boldsymbol{S}^3$, rank 14.

(f) Minimised $\hat{\boldsymbol{S}}^3$, rank 12.

Figure 4.4: Non-zero patterns of stoichiometric matrices $\boldsymbol{S}^l$ and $\hat{\boldsymbol{S}}^l$, for $l = 1,2,3$ derived from layering by flux and Flux Layer Minimisation respectively.

chapter allows the application of layered decomposition to a range of BRNs that allow transient (but not steady-state) mass flow between subsystems. More work is needed to characterise the types of BRNs in which a minimal layering by flux is or is not possible. For example, it is interesting that neither the gene transcription network (Example 4.1) nor the metabolic network (Section 4.5.2) admit a minimal layering by flux; however, a minimal layering of the example in Section 4.5.1 decomposes the system into ‘enzyme production’ and ‘enzyme action’ layers. Furthermore, the EFM constraint could also be used to inform the design of new, synthetic layers that can be integrated with an existing system such that the layered decomposition remains minimal.

In addition to the Flux Layer Minimisation approach, other formulations of the layered dynamics may exist that enforce other desirable properties of the decomposed system. For example, we can enforce the layered dynamics to be a cascade by re-writing the layered dynamics (4.7), assuming $L = 2$, as

$$\begin{aligned}\dot{\boldsymbol{y}}^1(t) &= \boldsymbol{S}^1 \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{y}^1(t)\big),\\ \dot{\boldsymbol{y}}^2(t) &= \big[\boldsymbol{S}^1 + \boldsymbol{S}^2\big]\, \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{y}^1(t) + \boldsymbol{y}^2(t)\big) - \boldsymbol{S}^1 \boldsymbol{v}\big(\boldsymbol{x}_0 + \boldsymbol{y}^1(t)\big).\end{aligned}$$

With a judicious choice of $\boldsymbol{S}^l$, this will enforce a cascade structure made up of stable, simpler layers. This approach is considered in much greater detail in [149], but will remain out of the scope of this thesis. Nevertheless, in that work we also consider an alternative interpretation of unstable layers. When the instability of one layer is balanced by the instability of another downstream layer, this is interpreted to mean that a perturbation to the downstream layer will cause the integrated system to ‘break’, in the sense that it will become unstable. This approach is based on the fact that, in the cascade formulation, the upstream layer must be independent of the downstream layer, and will thus remain unstable. Therefore, we can use a non-minimal layering to identify subsystems (and their corresponding parameters) which are necessary for the system’s stability.

The remainder of this thesis will go into further detail about how to choose and exploit layered decompositions, focusing on the partition of the reaction set $\mathscr{R}$ and constructing layered stoichiometric matrices using the Flux Layer Minimisation algorithm. In particular, we need to consider how to partition $\mathscr{R}$ into subsets that give biologically interpretable layers. The following chapter considers one approach which is particularly natural for many BRN models. The subsequent chapter considers the layered interconnection structure in more detail, and investigates layering strategies that optimise this structure in some sense. Finally, Chapter 7 will demonstrate the application of layered decomposition to the analysis of a large-scale BRN.

Chapter 5

Layering by Timescale

The previous chapter introduced the layered framework for the decomposition of biomolecular reaction networks, where it is assumed that the decomposition is defined by a partition of either the set $\mathscr{S}$ of species or the set $\mathscr{R}$ of reactions. A fundamental question for a layered decomposition is concerned with how the partition of $\mathscr{S}$ or $\mathscr{R}$ should be chosen. This chapter proposes a method for layering by flux based on timescale separation between reaction rates. In this case, the layered decomposition technique provides a new method of producing a quasi-steady state reduction of BRN models that no longer requires the species, or any combination of species, to be identified as fast or slow; the singularly perturbed dynamics can be directly written as a perturbation of the slow layer's dynamics. Furthermore, the layered framework allows the interactions between dynamics at multiple scales to be explicitly quantified.

5.1 Timescale Separation in Reaction Networks

The review in Section 1.2 and the motivation for the work in Chapter 3 was model order reduction, and in particular the singular perturbation of timescale-separated BRNs. This approach is underpinned by Tikhonov's Theorem, stated in Chapter 1 as Theorem 1.1. This theorem assumes that a small parameter $\mu \ll 1$ can be identified that characterises a separation in timescale between two components of a given dynamical system, expressed in the form (1.5). In this case, Tikhonov's Theorem implies that the reduced model (1.6) will be a good approximation to (1.5), in the limiting case as $\mu \to 0$. This approximation is known as the quasi-steady state (QSS) assumption. Chapter 3 is concerned with the quality of this and other approximations, and introduces a computational method for estimating the difference between the output trajectories of the original and reduced models.

The QSS approach, underpinned by Tikhonov's Theorem, is commonly used in mathematical biology [133] because timescale separation is a characteristic property of many biological systems [99, 116, 137]. A familiar example of QSS approximation is in the context of enzyme kinetics [28, 48, 205]. Section 3.2 reviews three QSS reduction methods to demonstrate the well-known fact that the validity of the assumptions made for each reduction strategy varies in parameter space. For example, the standard QSS approximation (sQSSA) given by (3.10) relies on the ratio E_0/X_0 between initial enzyme E_0 and initial substrate X_0 being very small. When this is not the case, the other types of QSS approximation become more valid.

When deriving the tQSS dynamics (3.12), it is assumed that the 'free substrate' concentration x cannot be classified as either fast or slow. Instead it is assumed that the combined 'free and bound substrate' concentration $\xi = x + c$ is slow, while the complex concentration c is fast. In this case, a transformation of the state space is required for the dynamics to be expressed in the form (1.5) necessary for Tikhonov's Theorem to apply.

Being unable to classify the timescale of x in this example is a particular case of the observation made in [116] that species can participate in both fast and slow processes; it is actually the reactions which can be considered as occurring on different timescales. The authors exploit their observation to motivate the design of a (possibly nonlinear) transformation of the state space into N new variables ξ_i, each of which can be classified as either fast or slow. Persisting with the enzyme kinetics example (3.7), suppose that the reversible binding $X + E \rightleftharpoons C$ is much faster than the production of the product $C \to Y + E$. In this case, C takes part in both fast and slow reactions, and thus the timescale of c is not well-defined. The observation that, in many cases, species concentrations do not separate in timescale has also been made in the context of stochastic simulations [36], and has motivated the method of prefactors, used to improve the quality of QSS approximations to gene transcription networks [23].

The notion that reactions separate in timescale is particularly suggestive in the context of the layered decomposition framework introduced in Chapter 4. If the species concentrations in a BRN can be classified as fast or slow variables, then the dynamics can be written in the form (1.5), which corresponds to a layering by species. If instead the reaction rates display timescale separation, then this naturally induces a partition of the reaction set $\mathscr{R}$. We will show below that layering by flux according to this partition provides the scale-separated dynamics required to apply Tikhonov's Theorem directly, resulting in an approximated system expressed in the same state space as the original system.

5.2 Layered QSS Approximation

5.2.1 Tikhonov's Theorem, layered by flux

Consider the standard ODE model (1.2) for a given BRN. Assuming that the reactions separate in timescale [23, 36, 116], consider the partition of the reaction set $\mathscr{R}$ into the sets $\mathscr{R}^s$ and $\mathscr{R}^f$ of slow and fast reactions respectively. Without loss of generality, it can be assumed that the reaction index is ordered so that the first M^s reactions are slow, and the next $M^f = M - M^s$ are fast. The scale separation of the last M^f reactions from the first M^s reactions implies that the flux vector $\boldsymbol{v}$ in (1.2) can be written as

$$\boldsymbol{v}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{v}^s(\boldsymbol{x}) \\ \boldsymbol{v}^f(\boldsymbol{x})/\mu \end{bmatrix},$$

for a small parameter $\mu \ll 1$. Here, we assume that the components $\boldsymbol{v}^s$ and $\boldsymbol{v}^f$ take values on the same scale as one another.

Consider the layering by flux, according to Definition 4.7, defined by the partition of $\mathscr{R}$ into the reaction subsets $\{\mathscr{R}^s, \mathscr{R}^f\}$. The stoichiometric matrix $\boldsymbol{S}$ is thus decomposed into the slow and fast stoichiometric matrices $\boldsymbol{S}^s$ and $\boldsymbol{S}^f$ respectively, with $\boldsymbol{S}^s + \boldsymbol{S}^f = \boldsymbol{S}$. Based on the decomposition of the flux vector above, define a re-scaled flux vector

$$\bar{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v}^s \\ \boldsymbol{v}^f \end{bmatrix}$$

such that all of the entries in $\bar{\boldsymbol{v}}$ are on the same scale. Then the layered dynamics (4.7) defined by $\boldsymbol{S}^s$ and $\boldsymbol{S}^f$ can be written as

$$\dot{\boldsymbol{x}}^s(t) = \boldsymbol{S}^s \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^s(t) + \boldsymbol{x}^f(t)\big), \tag{5.1a}$$

$$\mu \dot{\boldsymbol{x}}^f(t) = \boldsymbol{S}^f \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^s(t) + \boldsymbol{x}^f(t)\big), \tag{5.1b}$$

where $\boldsymbol{x}(0) = \boldsymbol{x}_0$ denotes the initial conditions of the original system, and where we set $\boldsymbol{x}^s(0) = \boldsymbol{x}^f(0) = \boldsymbol{0}$. The system state is retrieved from the layered decomposition through $\boldsymbol{x}(t) = \boldsymbol{x}_0 + \boldsymbol{x}^s(t) + \boldsymbol{x}^f(t)$.

The layered decomposition (5.1) defines a slow variable $\boldsymbol{x}^s$ and a fast variable $\boldsymbol{x}^f$ that are expressed in the scale-separated form (1.5) without any transformation of the state space. The aggregated dynamics $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{x}^s + \boldsymbol{x}^f$ imply the interpretation of each layer's state $\boldsymbol{x}^s(t)$ and $\boldsymbol{x}^f(t)$ as quantifying the contribution of the slow and fast reactions, respectively, to the overall system dynamics.

In [116], the authors also consider a layered approach, by constructing fast and slow variables they call $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$, that sum to give the exact state. The construction of these variables is different, however: for a projection $\mathscr{P}$ mapping onto the null-space of $\boldsymbol{S}^f$, they define $\boldsymbol{\xi} = \mathscr{P}\boldsymbol{x}$. This definition leads to slow dynamics for $\boldsymbol{\xi}$, but the dynamics of $\boldsymbol{\eta}$ are both slow and fast, and must be further approximated to leading order only. In contrast, the layered approach given here separates out the dynamics into 'true' fast and slow variables $\boldsymbol{x}^f$ and $\boldsymbol{x}^s$. Furthermore, the method in [116] requires the underlying reaction graph to be acyclic and have deficiency zero [79]. Our method is unconcerned with the CRNT approach, although it may be interesting to consider the relationship between this requirement and the assumed existence of the matrix inverse in $\boldsymbol{M}^f$.

We now return to the layered case. Having written the timescale-separated ODE system in the form (5.1), Tikhonov's Theorem implies that, as $\mu \to 0$, the solution to (5.1) is arbitrarily closely approximated by the solution to

$$\dot{\tilde{\boldsymbol{x}}}^s(t) = \boldsymbol{S}^s \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \tilde{\boldsymbol{x}}^f(t)\big), \tag{5.2a}$$

$$\boldsymbol{0} = \boldsymbol{S}^f \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \tilde{\boldsymbol{x}}^f(t)\big), \tag{5.2b}$$

where the approximated layers' states $\tilde{\boldsymbol{x}}^s$ and $\tilde{\boldsymbol{x}}^f$ sum to give $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \tilde{\boldsymbol{x}}^f(t)$. The initial condition of this system is at $\tilde{\boldsymbol{x}}^s(0) = \boldsymbol{0}$, as in the non-reduced case. However, in the approximated system (5.2) the fast state $\tilde{\boldsymbol{x}}^f$ is assumed to be at quasi-steady state, and hence to depend statically on the value of $\tilde{\boldsymbol{x}}^s$ according to the QSS manifold defined by the algebraic equation (5.2b). For this reason, the initial value of $\tilde{\boldsymbol{x}}^f(0)$ is no longer necessarily equal to zero, but instead determined by the equation $\boldsymbol{0} = \boldsymbol{S}^f \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{0} + \tilde{\boldsymbol{x}}^f(0)\big)$ given by the QSS manifold. The following section determines the conditions for which the QSS manifold provides a well-defined value of $\tilde{\boldsymbol{x}}^f(t)$ for any given $\tilde{\boldsymbol{x}}^s(t)$.

5.2.2 The QSS manifold

Section 4.3.1 made use of the decomposition of the layered stoichiometric matrices $\boldsymbol{S}^l = \boldsymbol{U}^l\boldsymbol{C}^l$ into the product of full-rank matrices consisting of $r^l = \operatorname{rank}(\boldsymbol{S}^l)$ columns and rows, respectively. Here, we recall this decomposition to further explore the QSS manifold defined by (5.2b).

Lemma 5.1. *The algebraic relation* (5.2b) *defining the QSS manifold is equivalent to the equation*

$$\boldsymbol{0} = \boldsymbol{C}^f\,\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \boldsymbol{U}^f\tilde{\boldsymbol{\xi}}^f(t)\big) \tag{5.3}$$

for the decomposition $\boldsymbol{S}^f = \boldsymbol{U}^f\boldsymbol{C}^f$ of the fast stoichiometric matrix, where $\tilde{\boldsymbol{x}}^f = \boldsymbol{U}^f\tilde{\boldsymbol{\xi}}^f$.

Proof. Similarly to the derivation of the minimal realisation (4.13) of the generic layered dynamics, consider the minimal realisation of the fast layer's dynamics (5.1b) given by

$$\mu\dot{\boldsymbol{\xi}}^f(t) = \boldsymbol{C}^f\,\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^s(t) + \boldsymbol{U}^f\boldsymbol{\xi}^f(t)\big),$$

for $\boldsymbol{x}^f = \boldsymbol{U}^f\boldsymbol{\xi}^f$, with initial conditions $\boldsymbol{\xi}^f(0) = \boldsymbol{0}$. The QSS manifold is found by letting $\mu \to 0$. If $\tilde{\cdot}$ denotes approximated variables, then this recovers (5.3), where the matrix $\boldsymbol{U}^f$ defines the output map $\tilde{\boldsymbol{x}}^f = \boldsymbol{U}^f\tilde{\boldsymbol{\xi}}^f$. □

The relationship (5.3) evaluated at $t = 0$ may not uniquely define the initial value of $\tilde{\boldsymbol{\xi}}^f(0)$. In this case, the 'correct' initial condition to choose is given by first isolating the fast layer, through setting $\boldsymbol{x}^s(t) \equiv \boldsymbol{0}$ in (5.1b), and then matching the timescales $\tilde{\boldsymbol{\xi}}^f(0) = \lim_{t\to\infty}\boldsymbol{\xi}^f(t)$, according to standard singular perturbation theory [133].

In order to apply Lemma 5.1, a function $\boldsymbol{F} : \mathbb{R}^N \times \mathbb{R}^{r^f} \to \mathbb{R}^{r^f}$ can be defined such that

$$\boldsymbol{F}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{C}^f\,\bar{\boldsymbol{v}}(\boldsymbol{\alpha} + \boldsymbol{U}^f\boldsymbol{\beta}). \tag{5.4}$$

With this definition, (5.3) is equivalent to $\boldsymbol{F}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t), \tilde{\boldsymbol{\xi}}^f(t)\big) = \boldsymbol{0}$. The following results depend on the following Implicit Function Theorem.

Theorem 5.2 (Implicit Function Theorem (IFT) [183])**.** *For a function $\boldsymbol{G} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$, consider the manifold in $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ defined by the algebraic equation $\boldsymbol{0} = \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{y})$ for $\boldsymbol{x} \in \mathbb{R}^{n_1}$ and $\boldsymbol{y} \in \mathbb{R}^{n_2}$. Denote by $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{y}}(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ the Jacobian of $\boldsymbol{G}$ with respect to $\boldsymbol{y}$, evaluated at a given point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$. If this $n_2 \times n_2$ square matrix is non-singular, then there exist open sets $\mathscr{D}_i \subseteq \mathbb{R}^{n_i}$, with $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in \mathscr{D}_1 \times \mathscr{D}_2$, and there exists a function $\boldsymbol{g} : \mathscr{D}_1 \to \mathscr{D}_2$, such that $\boldsymbol{G}\big(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x})\big) = \boldsymbol{0}$ for all $\boldsymbol{x} \in \mathscr{D}_1$.*

Furthermore, if these conditions hold, the Jacobian of $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is given by the formula

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} = -\left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{y}}\right)^{-1}\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{x}},$$

evaluated at $\big(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x})\big)$.

The IFT can be used to determine how the contribution $\tilde{\boldsymbol{x}}^f(t)$ of the approximated fast layer depends on the contribution $\tilde{\boldsymbol{x}}^s(t)$ of the slow layer. In the following, the notation $\bar{\boldsymbol{v}}'(\boldsymbol{x})$ is used as shorthand to denote the Jacobian of $\bar{\boldsymbol{v}}$ evaluated at $\boldsymbol{x}$.

Proposition 5.3. *If the $r^f \times r^f$ square matrix $\boldsymbol{C}^f \bar{\boldsymbol{v}}'(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f)\boldsymbol{U}^f$ is invertible, then there exists a function $\boldsymbol{\phi} : \mathbb{R}^N \to \mathbb{R}^N$ such that $\tilde{\boldsymbol{x}}^f = \boldsymbol{\phi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)$ satisfies* (5.2b).

Corollary 5.4. *Assume that the condition of Proposition 5.3 is met. Then the Jacobian of $\tilde{\boldsymbol{x}}^f = \boldsymbol{\phi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)$ with respect to $\tilde{\boldsymbol{x}}^s$ is*

$$\begin{aligned}\frac{\mathrm{d}\tilde{\boldsymbol{x}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s} &= -\boldsymbol{U}^f \left[\boldsymbol{C}^f \bar{\boldsymbol{v}}'(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f)\boldsymbol{U}^f\right]^{-1} \boldsymbol{C}^f \bar{\boldsymbol{v}}'(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f) \\ &=: \boldsymbol{M}^f(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f), \end{aligned} \tag{5.5}$$

where $\frac{\mathrm{d}\tilde{\boldsymbol{x}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s}$ denotes the Jacobian of $\tilde{\boldsymbol{x}}^f$ with respect to $\tilde{\boldsymbol{x}}^s$. The definition of $\boldsymbol{M}^f$ is independent of the bases of $\mathrm{Col}(\boldsymbol{S}^f)$ *and* $\mathrm{Row}(\boldsymbol{S}^f)$ *used to construct $\boldsymbol{U}^f$ and $\boldsymbol{C}^f$ respectively.*

Proof. These results follow immediately from applying the IFT to the function $\boldsymbol{F}$ in (5.4), where Lemma 5.1 implies that (5.2b) is equivalent to $\boldsymbol{F}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}^s}, \tilde{\boldsymbol{\xi}}^f) = \boldsymbol{0}$. The Jacobian of $\boldsymbol{F}$ with respect to $\tilde{\boldsymbol{\xi}}^f$ is

$$\frac{\partial \boldsymbol{F}}{\partial \tilde{\boldsymbol{\xi}}^f} = \boldsymbol{C}^f \bar{\boldsymbol{v}}'(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \boldsymbol{U}^f \tilde{\boldsymbol{\xi}}^f)\boldsymbol{U}^f.$$

By Theorem 5.2, if this $r^f \times r^f$ square matrix evaluated at the given $(\tilde{\boldsymbol{x}}^s, \tilde{\boldsymbol{x}}^f)$ is invertible then a function $\boldsymbol{\psi} : \mathbb{R}^N \to \mathbb{R}^{r^f}$ exists such that $\tilde{\boldsymbol{\xi}}^f = \boldsymbol{\psi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)$ locally around $\tilde{\boldsymbol{x}}^s$. The value of $\tilde{\boldsymbol{x}}^f = \boldsymbol{U}^f \tilde{\boldsymbol{\xi}}^f$ can then be found by composing $\tilde{\boldsymbol{x}}^f = \boldsymbol{\phi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s) := \boldsymbol{U}^f \boldsymbol{\psi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)$. Clearly this function satisfies (5.2b), and hence Proposition 5.3 holds.

To prove Corollary 5.4, the second part of Theorem 5.2 implies that the function $\tilde{\boldsymbol{\xi}}^f = \boldsymbol{\psi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)$ has derivative

$$\begin{aligned}\frac{\mathrm{d}\tilde{\boldsymbol{\xi}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s} &= -\left[\frac{\partial \boldsymbol{F}}{\partial \tilde{\boldsymbol{\xi}}^f}\right]^{-1} \frac{\partial \boldsymbol{F}}{\partial \tilde{\boldsymbol{x}}^s} \\ &= -\left[\boldsymbol{C}^f \bar{\boldsymbol{v}}'\left(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \boldsymbol{U}^f \tilde{\boldsymbol{\xi}}^f\right)\boldsymbol{U}^f\right]^{-1} \boldsymbol{C}^f \bar{\boldsymbol{v}}'\left(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \boldsymbol{U}^f \tilde{\boldsymbol{\xi}}^f\right).\end{aligned}$$

Since $\tilde{\boldsymbol{x}}^f = \boldsymbol{U}^f \tilde{\boldsymbol{\xi}}^f$, this derivative can be pre-multiplied by $\boldsymbol{U}^f$ to give

$$\frac{\mathrm{d}\tilde{\boldsymbol{x}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s} = \boldsymbol{U}^f \frac{\mathrm{d}\tilde{\boldsymbol{\xi}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s} = \boldsymbol{M}^f(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f),$$

thus proving (5.5).

Finally, consider another decomposition of $\boldsymbol{S}^f = \bar{\boldsymbol{U}}^f \bar{\boldsymbol{C}}^f = \boldsymbol{U}^f \boldsymbol{C}^f$. These matrices can be related by the invertible $r^f \times r^f$ transformation $\boldsymbol{T}$, representing a change of basis, such that $\bar{\boldsymbol{U}}^f = \boldsymbol{U}^f \boldsymbol{T}$ and $\bar{\boldsymbol{C}}^f = \boldsymbol{T}^{-1}\boldsymbol{C}^f$. Forming $\bar{\boldsymbol{M}}^f$ using $\bar{\boldsymbol{U}}^f$ and $\bar{\boldsymbol{C}}^f$ according to the formula given in (5.5) we find that

$$\begin{aligned}\bar{\boldsymbol{M}}^f &= -\bar{\boldsymbol{U}}^f \left[\bar{\boldsymbol{C}}^f \bar{\boldsymbol{v}}' \bar{\boldsymbol{U}}^f\right]^{-1} \bar{\boldsymbol{C}}^f \bar{\boldsymbol{v}}' \\ &= -\boldsymbol{U}^f \boldsymbol{T} \left[\boldsymbol{T}^{-1}\boldsymbol{C}^f \bar{\boldsymbol{v}}' \boldsymbol{U}^f \boldsymbol{T}\right]^{-1} \boldsymbol{T}^{-1}\boldsymbol{C}^f \bar{\boldsymbol{v}}' \\ &= -\boldsymbol{U}^f \boldsymbol{T}\boldsymbol{T}^{-1} \left[\boldsymbol{C}^f \bar{\boldsymbol{v}}' \boldsymbol{U}^f\right]^{-1} \boldsymbol{T}\boldsymbol{T}^{-1}\boldsymbol{C}^f \bar{\boldsymbol{v}}' \\ &= \boldsymbol{M}^f,\end{aligned}$$

which implies that $\boldsymbol{M}^f$ is invariant to the choice of decomposition $\boldsymbol{S}^f = \boldsymbol{U}^f \boldsymbol{C}^f$. □

If both the fast layer's stoichiometric matrix $\boldsymbol{S}^f = \boldsymbol{U}^f \boldsymbol{C}^f$ and the re-scaled flux vector $\bar{\boldsymbol{v}}$ satisfy the hypothesis of Proposition 5.3, then this result implies that there exists a function $\boldsymbol{\phi}$ such that the approximated dynamics (5.2) can be written as

$$\dot{\tilde{\boldsymbol{x}}}^s(t) = \boldsymbol{S}^f \bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \tilde{\boldsymbol{x}}^f(t)\big), \tag{5.6a}$$

$$\tilde{\boldsymbol{x}}^f(t) = \boldsymbol{\phi}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t)\big), \tag{5.6b}$$

with initial conditions $\tilde{\boldsymbol{x}}^s = \boldsymbol{0}$, and such that the approximated state of the timescale-separated system is given by $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \boldsymbol{\phi}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t)\big)$.

As explained in the previous chapter, the dimension of the ODE (5.6a) is equal to $r^s = \text{rank}(\boldsymbol{S}^s)$. Assuming that $r^s < r = \text{rank}(\boldsymbol{S})$, this therefore defines a model reduction. However, a key problem with this method is the explicit calculation of the function $\boldsymbol{\phi}$ implied by the manifold (5.3). Corollary 5.4 gives some indication of how this function depends on $\tilde{\boldsymbol{x}}^s$. In the following section, we will define an alternative representation of the approximated, reduced dynamics (5.6) that obviates the requirement to find $\boldsymbol{\phi}(\tilde{\boldsymbol{x}}^s)$, except at the initial conditions.

5.2.3 Timescales and layered interconnection structure

The dynamics (5.6) result in approximated layer trajectories $\tilde{\boldsymbol{x}}^s(t)$ and $\tilde{\boldsymbol{x}}^f(t)$ that sum to give the approximated system trajectory $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t) + \boldsymbol{\phi}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s(t)\big)$. We now use Corollary 5.4 to express the dynamics of the approximated system state $\tilde{\boldsymbol{x}}(t)$.

Theorem 5.5. *Consider the timescale-separated layered decomposition of the system with fast and slow stoichiometric matrices $\boldsymbol{S}^f$ and $\boldsymbol{S}^s$ respectively corresponding to the fast and slow reactions, given by the layered dynamics* (5.1)*, where $\mu \ll 1$ parametrises the speed of the fast reactions. Then, as $\mu \to 0$, the solution $\tilde{\boldsymbol{x}}$ to*

$$\dot{\tilde{\boldsymbol{x}}} = \Big(\boldsymbol{I} + \boldsymbol{M}^f(\tilde{\boldsymbol{x}})\Big)\boldsymbol{S}^s \bar{\boldsymbol{v}}(\tilde{\boldsymbol{x}}), \tag{5.7}$$

where $\boldsymbol{M}^f$ is defined in (5.5)*, arbitrarily closely approximates the solution $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{x}^f + \boldsymbol{x}^s$ of the full system* (5.1)*. The initial conditions of* (5.7) *are equal to $\tilde{\boldsymbol{x}}(0) = \boldsymbol{x}_0 + \boldsymbol{\phi}(\boldsymbol{x}_0)$, where $\boldsymbol{\phi}(\boldsymbol{x}_0) = \lim_{t\to\infty} \hat{\boldsymbol{x}}^f(t)$ is the steady state of the isolated fast layer with dynamics $\dot{\hat{\boldsymbol{x}}}^f = \boldsymbol{S}^f \boldsymbol{v}(\boldsymbol{x}_0 + \hat{\boldsymbol{x}}^f)$, from $\hat{\boldsymbol{x}}^f(0) = \boldsymbol{0}$.*

Proof. Tikhonov's Theorem can be applied to conclude that (5.2), and therefore (5.6), is an arbitrarily close approximation to the system dynamics (5.1) as $\mu \to 0$. It remains to show that $\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \tilde{\boldsymbol{x}}^f$, given by the solutions of (5.6), is equal to $\tilde{\boldsymbol{x}}$ given by the solution to (5.7).

By taking the derivative of the former, we find that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \boldsymbol{\phi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)\big) &= \left(\dot{\tilde{\boldsymbol{x}}}^s + \frac{\mathrm{d}\tilde{\boldsymbol{x}}^f}{\mathrm{d}\tilde{\boldsymbol{x}}^s}\dot{\tilde{\boldsymbol{x}}}^s\right) \\ &= \Big(\boldsymbol{I} + \boldsymbol{M}^f\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s + \boldsymbol{\phi}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^s)\big)\Big)\dot{\tilde{\boldsymbol{x}}}^s,\end{aligned}$$

matching the form in (5.7). Furthermore, the initial conditions $\tilde{\boldsymbol{x}}(0)$ of (5.7) are assumed to be equal to the aggregated initial conditions $\boldsymbol{x}_0 + \boldsymbol{0} + \boldsymbol{\phi}(\boldsymbol{x}_0)$ of the layered system (5.6). Hence the two trajectories are equal, and thus (5.7) is a non-stiff QSS approximation to (5.1). □

Theorem 5.5 proves that the QSS approximation to the timescale-separated layered system (5.1) is given by the dynamics (5.7). The definition of the matrix $\boldsymbol{M}^f$ given in (5.5) implies that $(\boldsymbol{M}^f)^2 = -(\boldsymbol{M})$, and hence that $(\boldsymbol{I}+\boldsymbol{M}^f)^2 = (\boldsymbol{I}+\boldsymbol{M}^f)$. This factor, which multiplies the slow layer's isolated dynamics $\boldsymbol{S}^s\bar{\boldsymbol{v}}$, is a projection. The QSS approximated dynamics (5.7) can therefore be considered as being in 'projection form'.

By writing the approximated dynamics in projection form, the effect of the fast layer on the slow layer can be characterised by the matrix $\boldsymbol{M}^f\big(\tilde{\boldsymbol{x}}(t)\big)$. For example, if $\boldsymbol{M}^f\boldsymbol{S}^s \approx \boldsymbol{0}$ then the fast layer has very little effect on the slow dynamics; the slow layer in isolation, ignoring the fast dynamics, is a reasonable approximation to the system. In other cases there may be input directions of Row$(\boldsymbol{M}^f)$ that are closely aligned to vectors in Col$(\boldsymbol{S}^s)$, implying that, through $\boldsymbol{M}^f$, the fast layer exerts a strong influence on the dynamics at the slow timescale. Furthermore, the dependence of the perturbation $\boldsymbol{M}^f\big(\tilde{\boldsymbol{x}}(t)\big)$ on the state implies that the strength of the fast layer's influence is dynamic. Thus, an analysis of the angles between the column space of $\boldsymbol{S}^s$ and the time-varying row space of $\boldsymbol{M}^f\big(\tilde{\boldsymbol{x}}(t)\big)$ will determine the strength of the fast layer's influence at the slow timescale.

Similarly to the original layered decomposition, the projection form (5.7) does not immediately appear to be a reduction of the layered system. In Section 4.3.1 it was observed that the dimension of a layer is defined by the rank of its stoichiometric matrix. From the equivalent dynamics (5.6) the dimension of the reduced system is clearly equal to $r^s = \text{rank}(\boldsymbol{S}^s)$. However, the time-varying projection $(\boldsymbol{I}+\boldsymbol{M}^f)$ pre-multiplying $\boldsymbol{S}^s\bar{\boldsymbol{v}}$ means that the dynamics of (5.7) are not expressed in the form $\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{v}}$. Therefore a minimal realisation of these dynamics in a state space of r^s variables is not as easy to determine as it is for (5.6a).

Nevertheless, in projection form the effect of the QSS approximation on the dimension of the system can be seen by pre-multiplying $\dot{\boldsymbol{x}}$ with $\boldsymbol{M}^f$. The fact that $(\boldsymbol{I}+\boldsymbol{M}^f)$ is a projection means that $\boldsymbol{M}^f(\tilde{\boldsymbol{x}})\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{0}$. Therefore $\dot{\tilde{\boldsymbol{x}}}$ is constrained to be orthogonal to the row space Row$\big(\boldsymbol{M}^f(\tilde{\boldsymbol{x}})\big)$, which is of dimension r^f. Hence there are r^f (potentially nonlinear) conservation constants satisfied by the reduced system (5.7).

A key benefit of the layered approach to QSS approximation is that it quickly produces reduced models without requiring transformations of the state space to be found, and therefore expresses the dynamics of an approximated, lower-dimensional system in the original system coordinates. Furthermore, the projection form (5.7) can be written directly from the fast layer's stoichiometric matrix and the Jacobian of the flux vector. Thus the implicit function defined by the algebraic equation (5.2b) defining the fast subsystem's QSS manifold does not need to be found explicitly. Only a single point $\tilde{\boldsymbol{x}}^f(0) = \boldsymbol{\phi}(\boldsymbol{x}_0)$ on this manifold needs to be determined, in order to define the initial condition $\tilde{\boldsymbol{x}}(0) = \boldsymbol{x}_0 + \boldsymbol{\phi}(\boldsymbol{x}_0)$.

5.2.4 Examples

The following two examples illustrate the benefits of applying the layered framework to model reduction by QSS approximation. The first example, based on a simple model of the molecular dynamics of a viral infection and originally given in [191], is a simple example that illustrates how the layered approach can be applied. The second example (returning to the enzyme kinetics BRN (3.7)) is intended to make the benefits of this approach much more apparent. The established approach of transforming the state space into fast and slow variables produces an ODE model that is far less clear than that produced by the new, direct approach of layering by flux according to the timescale-separated reactions.

5.2.4.1 Intracellular viral infection

To demonstrate the layered approach to timescale separation, consider the following model of an intracellular viral infection used as Example 4 in [116]. The complete set of reactions is taken from [191], describing the positive feedback of a viral genome D by its product R, which also goes on to produce additional viral protein P. The viral protein and viral DNA combine to result in additional synthesised virus V. These reactions are given by

$$\begin{aligned} &D \xrightarrow{k_1} R + D, && R \xrightarrow{k_2} \emptyset, \\ &R \xrightarrow{k_3} R + D, && D + P \xrightarrow{k_4} V, \\ &R \xrightarrow{k_5/\mu} P + R, && P \xrightarrow{k_6/\mu} \emptyset, \end{aligned}$$

where protein production and degradation, modelled by the final two reactions, are assumed to occur on a much faster timescale than the other processes. Hence P takes part in both fast and slow reactions, so that its concentration cannot be considered as being either a fast or slow variable[1]. By denoting $\boldsymbol{x} = \big([D],[R],[V],[P]\big)^T$ as the concentration vector, this BRN can be modelled as a rank($\boldsymbol{S}$) = 4-dimensional ODE system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x_1 \\ k_2 x_2 \\ k_3 x_2 \\ k_4 x_1 x_4 \\ k_5 x_2/\mu \\ k_6 x_4/\mu \end{bmatrix}, \tag{5.8}$$

for the small parameter $\mu \ll 1$, assuming initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0 = (x_{1,0},\ldots,x_{4,0})^T$.

This system can be approximated by applying Theorem 5.5. Partitioning the reaction set into $\mathscr{R}^s = \{R_1,\ldots,R_4\}$ and $\mathscr{R}^f = \{R_5,R_6\}$ defines a layering by flux such that

$$\boldsymbol{S}^s + \boldsymbol{S}^f = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

[1] There is no net production or consumption of R in the fast reaction 5, so that its concentration can still be considered a slow variable.

According to Theorem 5.5, the dynamics of the QSS approximation to this system are

$$\dot{\tilde{\boldsymbol{x}}} = \left(\boldsymbol{I} + \boldsymbol{M}^f\left(\tilde{\boldsymbol{x}}(t)\right)\right) \begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} k_1\tilde{x}_1 \\ k_2\tilde{x}_2 \\ k_3\tilde{x}_2 \\ k_4\tilde{x}_1\tilde{x}_4 \end{bmatrix},$$

where the perturbation $\boldsymbol{M}^f$ is given by substituting the matrices

$$\boldsymbol{U}^f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{C}^f = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \bar{\boldsymbol{v}}' = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & k_3 & 0 & 0 \\ k_4x_4 & 0 & 0 & k_4x_1 \\ 0 & k_5 & 0 & 0 \\ 0 & 0 & 0 & k_6 \end{bmatrix}$$

into the formula (5.5), resulting in the perturbation

$$\boldsymbol{M}^f = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & k_5/k_6 & 0 & -1 \end{bmatrix}$$

and hence the approximated dynamics in projection form (5.7) of

$$\dot{\tilde{\boldsymbol{x}}} = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_5}{k_6} & -\frac{k_5}{k_6} & 0 & 0 \end{bmatrix} \begin{bmatrix} k_1\tilde{x}_1 \\ k_2\tilde{x}_2 \\ k_3\tilde{x}_2 \\ k_4\tilde{x}_1\tilde{x}_4 \end{bmatrix}. \tag{5.9}$$

The initial conditions of the approximated system are equal to $\boldsymbol{x}_0 + \boldsymbol{\phi}(\boldsymbol{x}_0)$, where the initial condition of the approximated fast layer $\boldsymbol{\phi}(\boldsymbol{x}_0) = \lim_{t\to\infty} \hat{\boldsymbol{x}}^f(t)$ is the steady state value of the fast layer in isolation. This is the system with dynamics

$$\dot{\hat{\boldsymbol{x}}}^f = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} k_5(x_{2,0} + x_2^f) \\ k_6(x_{4,0} + x_4^f) \end{bmatrix},$$

initialised from $\hat{\boldsymbol{x}}^f = \boldsymbol{0}$. It is trivial to find $\boldsymbol{\phi}(\boldsymbol{0}) = \left(0, 0, 0, \frac{k_5}{k_6}x_{2,0} - x_{4,0}\right)^T$.

The new stoichiometric matrix in (5.9) is constant and of rank 3. Therefore the projection form (5.7), in this case at least, produces a reduced-dimensional system compared to the original 4-dimensional system (5.8). The ratio k_5/k_6 determines the extent to which the fast dynamics influence the isolated slow dynamics, quantified by the angle between the row space of $\boldsymbol{M}^f$ and the column space of $\boldsymbol{S}^s$. The smaller k_5/k_6 is, the greater the effect of the fast reactions on the slow dynamics. As k_5/k_6 grows larger, the row space of $\boldsymbol{M}^f$ becomes more orthogonal to the column space of $\boldsymbol{S}^s$, and the effect of the fast reactions is reduced.

The equation $\boldsymbol{M}^f(\tilde{\boldsymbol{x}})\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{0}$ implies that the vector field of the reduced system is always orthogonal to the row space of $\boldsymbol{M}^f$, spanned by the vector $[0, k_5/k_6, 0, -1]^T$. Hence we can conclude that $\frac{k_5}{k_6}\dot{\tilde{x}}_2 - \dot{\tilde{x}}_4 = 0$, and thus that $k_5\tilde{x}_2 - k_6\tilde{x}_4$ is constant. Indeed, from the initial conditions $\tilde{\boldsymbol{x}}(0)$ of the approximated system we can conclude that $k_5\tilde{x}_2(t) - k_6\tilde{x}_4(t) = 0$ for all t.

The conclusions of this example are not difficult to achieve through the more established methods for model reduction. In this case, we would define a new fast variable $z = x_3 + x_4$, which is assumed to reach quasi-steady state. Nevertheless, now that the layered approach has been demonstrated for the QSS reduction of this simple nonlinear system, the following example is intended to illustrate the benefits of the layered approach. In particular, the approximated system will be expressed in the original state space, meaning that no state-space transformation is required, and ensuring that the approximated system can be analysed in terms of the originally-modelled quantities.

5.2.4.2 Enzyme kinetics

Consider again the enzyme kinetics introduced in Section 3.2, and now assume that the reversible binding of the substrate X with enzyme E is fast. The reaction rates can be parametrised by a small positive constant $\mu \ll 1$ such that

$$X + E \underset{k_{-1}/\mu}{\overset{k_1/\mu}{\rightleftharpoons}} C \xrightarrow{k_2} Y + E,$$

with k_{-1}, k_1, and k_2 on the same scale. The dynamics of this BRN are given by (3.8) with scaled reaction rates, such that

$$\begin{bmatrix} \dot{x}(t) \\ \dot{c}(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} k_1 x(t)\,[E_0 - c(t)]/\mu \\ k_{-1}c(t)/\mu \\ k_2 c(t) \end{bmatrix}, \tag{5.10}$$

with initial conditions $\big(x(0), c(0)\big) = (x_0, c_0)$. This ODE implies that, while x is a fast variable, the variable c cannot be classified as either fast or slow, since it takes part in both fast and slow reactions.

The typical strategy for timescale analysis in this case is to transform the state space, so that the new variable $z = x + c$ will be slow, while x is fast. The QSS approximation of the transformed system results in approximated slow dynamics $\dot{\tilde{z}} = -k_2(\tilde{z} - \tilde{x})$, with $\tilde{x}$ found in terms of $\tilde{z}$ by the solution to the following algebraic equation

$$k_1\tilde{x}^2 - (k_1\tilde{z} + k_1E_0 + k_{-1})\tilde{x} - k_{-1}\tilde{z} = 0,$$

defining the QSS manifold. Hence the resulting approximated dynamics are not expressed directly in the concentrations x and c originally being modelled. Furthermore, the resulting ODE, written in terms of $\tilde{z}$ as

$$\dot{\tilde{z}} = -k_2 \left(\tilde{z} - \frac{k_1 z + k_1 E_0 + k_{-1} + \sqrt{(k_1 z + k_1 E_0 + k_{-1})^2 + 4 k_1 k_{-1}}}{2k_1} \right) \tag{5.11}$$

is not easily interpreted, biochemically.

We can now compare this approach with the direct, layered, reduction method introduced in this chapter. Indexing the three reactions according to the columns of $\boldsymbol{S}$ written above, the reactions partition into $\mathscr{R}^f = \{R_1, R_2\}$ and $\mathscr{R}^s = \{R_3\}$. This implies that the corresponding layered stoichiometric matrices are

$$\boldsymbol{S}^f + \boldsymbol{S}^s = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Furthermore, the flux vector $\boldsymbol{v}$ is rescaled into $\bar{\boldsymbol{v}}^T = [k_1 x(E_0 - c), k_{-1} c, k_2 c]$. Applying Theorem 5.5, the approximated dynamics are equal to

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{c}} \end{bmatrix} = \left(\boldsymbol{I} + \boldsymbol{M}^f(\tilde{x}, \tilde{c}) \right) \begin{bmatrix} 0 \\ -1 \end{bmatrix} k_2 \tilde{c},$$

where $\boldsymbol{M}^f(\tilde{x}, \tilde{c})$ can be found by the formula (5.5) in terms of the column and row spaces of the fast layer's stoichiometric matrix $\boldsymbol{S}^f$ and the Jacobian of $\bar{\boldsymbol{v}}$. Substituting the matrices

$$\boldsymbol{U}^f = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \ \boldsymbol{C}^f = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}, \ \bar{\boldsymbol{v}}' = \begin{bmatrix} k_1(E_0 - c) & -k_1 x \\ 0 & k_{-1} \\ 0 & k_2 \end{bmatrix}$$

into (5.5) implies that the matrix $\boldsymbol{M}^f$ defining the approximated dynamics in projection form (5.7) is given by

$$\boldsymbol{M}^f(\tilde{x}, \tilde{c}) = \frac{1}{k_1(E_0 - \tilde{c}) + (k_1 \tilde{x} + k_{-1})} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} k_1(E_0 - \tilde{c}) & -(k_1 \tilde{x} + k_{-1}) \end{bmatrix}.$$

By denoting $A(\tilde{x}, \tilde{c}) = k_1(E_0 - \tilde{c})$ and $B(\tilde{x}, \tilde{c}) = k_1 \tilde{x} + k_{-1}$, this matrix can then be substituted into (5.7) to give the QSS approximated dynamics

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{c}} \end{bmatrix} = \frac{1}{A+B} \begin{bmatrix} B & B \\ A & A \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} k_2 \tilde{c} = \frac{-1}{A+B} \begin{bmatrix} B \\ A \end{bmatrix} k_2 \tilde{c} \tag{5.12}$$

from initial conditions $\left(\tilde{x}(0), \tilde{c}(0)\right)^T = (x_0, c_0)^T + \boldsymbol{\phi}(x_0, c_0)$. Here, the fast layer's initial condition $\boldsymbol{\phi}(x_0, c_0) = \lim_{t \to \infty} \left(\hat{x}^f(t), \hat{c}^f(t)\right)^T$ is the steady state of the isolated fast layer

$$\begin{bmatrix} \dot{\hat{x}}^f \\ \dot{\hat{c}}^f \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} k_1 \left(x_0 + \hat{x}^f\right)\left(E_0 - c_0 - \hat{c}^f\right) \\ k_{-1}\left(c_0 + \hat{c}^f\right) \end{bmatrix},$$

initialised from $\left(\hat{x}^f(0), \hat{c}^f(0)\right)^T = \boldsymbol{0}$, which can either be found numerically or (in this case) analytically by solving a single quadratic equation.

Clearly, the ODE (5.12) modelling the QSS approximated dynamics quantifies the effect of the fast layer on the slow dynamics much more explicitly than the result (5.11) of the more established approach. In particular, the slow layer's dynamics in isolation model the complex concentration's rate of decay by $\dot{c}^s = -k_2(c_0 + c^s)$, starting from $c^s(0) = 0$. However, the effect of the fast layer is to multiply this decay rate by a factor $\frac{A}{A+B} \leq 1$, thereby slowing the decay. As C is lost by the production reaction $C \rightarrow P + E$, it is partly replenished by the fast reactions that instantly equilibriate X and C. Thus the decrease in C is slower.

The extent of the fast reactions' effect on the slow timescale can be altered by adapting the fast layer's parameters. In particular, if $E_0 \ll 1$, then the row space of $\text{Row}(\boldsymbol{M}^f)$ is generated by the single vector $(A, -B)$ that is approximately parallel to $[0, -1]$. Hence the row space $\text{Row}(\boldsymbol{M}^f)$ is closely aligned with the column space of $\boldsymbol{S}^s$, implying that the fast layer has a very strong impact on the slow dynamics, slowing the decay of c. Conversely, for large amounts of total enzyme $E_0 \gg k_{\pm 1}$, the vector $(A, -B)$ generating the row space $\text{Row}(\boldsymbol{M}^f)$ is now closely aligned with $[1, 0]$, and therefore approximately orthogonal to the slow dynamics. In this case, the effect of the fast reactions on the slow dynamics is greatly reduced.

As mentioned in Section 5.2.3, the fact that the dynamics (5.12) are a reduction of (5.10) is not immediately apparent. Although the stiffness has been approximated, there are still two coordinates, and there is no obvious small-rank stoichiometry that we can use to express the dynamics in a lower-dimension state space, in the manner described in Section 4.3.1. However, the relationship $\boldsymbol{M}(\tilde{\boldsymbol{x}})\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{0}$ implies, in this case, that

$$k_1(E_0 - \tilde{c})\dot{\tilde{x}} - (k_1\tilde{x} + k_{-1})\dot{\tilde{c}} = 0,$$

or equivalently that $k_1(E_0 - \tilde{c})\tilde{x} - k_{-1}\tilde{c}$ is constant. The value of this constant is found by substituting the initial conditions $\big(\tilde{x}(0), \tilde{c}(0)\big)$. The approximated solution remains on this nonlinear manifold in phase space, reducing the system from two to one dimension.

This example has demonstrated that the layered framework, applied to QSS approximation, can produce a reduced, approximated system directly in terms of the original system coordinates. The approximated dynamics are easily expressed in terms of the stoichiometry and Jacobian of the fast reactions, and can be interpreted in terms of the interaction between the fast and slow subsystems. The following section develops the approach introduced here to apply to systems that separate into more than two timescales.

5.3 Multiple Timescales

Large scale biological models may contain so many processes that they can be separated into more than 2 timescales. Suppose that the reaction set $\mathscr{R}$ admits a partition into $L > 2$ disjoint subsets $\mathscr{R}^l$, for $l = 1, \ldots, L$, where each subset corresponds to a distinct timescale. The layered approach to timescale separation remains applicable in this case, by simply layering by flux into L layers with stoichiometric matrices $\boldsymbol{S}^l$ corresponding to the reaction subsets $\mathscr{R}^l$. Assume that the reaction index is ordered such that the flux vector $\boldsymbol{v}$ can be written to model the L timescales by

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}^1 \\ \boldsymbol{v}^2/\mu \\ \vdots \\ \boldsymbol{v}^L/(\mu)^{L-1} \end{bmatrix},$$

where the gap between successive timescales is parametrised by the small positive constant $\mu \ll 1$. The components of each $\boldsymbol{v}^i$ are assumed to take values on the same scale.

Similarly to the derivation of (5.1), a rescaled flux vector $\bar{\boldsymbol{v}} = \left[\boldsymbol{v}^1,\dots,\boldsymbol{v}^L\right]^T$ can be used together with the layered stoichiometric matrices $\boldsymbol{S}^l$ to define the layered dynamics

$$(\mu)^{l-1}\dot{\boldsymbol{x}}^l(t) = \boldsymbol{S}^l\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^1(t) + \cdots + \boldsymbol{x}^L(t)\big), \tag{5.13}$$

with initial conditions $\boldsymbol{x}^l(0) = \boldsymbol{0}$ for each $l = 1,\dots,L$. These dynamics are a realisation of the timescale-separated ODE (1.2), which can now be approximated using the layered QSS technique introduced in Section 5.2.

5.3.1 Layered QSS approximation

First, a timescale of interest needs to be selected. Without loss of generality, we can assume that this timescale is the slowest timescale in the model. This assumption can be justified by first considering any faster timescale, corresponding to a layer k where $1 < k \leq L$. In order to approximate the system at this timescale, t needs to be rescaled by a factor of $(\mu)^{k-1}$ so that the layered dynamics (5.13) are re-written

$$\begin{aligned} \dot{\boldsymbol{x}}^l(t) &= (\mu)^{k-l}\boldsymbol{S}^l\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^1(t) + \cdots + \boldsymbol{x}^L(t)\big) \text{ for } l = 1,\dots,(k-1), \\ \dot{\boldsymbol{x}}^k(t) &= \boldsymbol{S}^k\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^1(t) + \cdots + \boldsymbol{x}^L(t)\big), \\ (\mu)^{l-k}\dot{\boldsymbol{x}}^l(t) &= \boldsymbol{S}^l\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \boldsymbol{x}^1(t) + \cdots + \boldsymbol{x}^L(t)\big) \text{ for } l = (k+1),\dots,L. \end{aligned}$$

As usual, the QSS approximation is to consider the system in the limit as $\mu \to 0$. In this case, the states $\boldsymbol{x}^l$ of layers $l = 1,\dots,(k-1)$ are approximated to be constant $\tilde{\boldsymbol{x}}^l(t) = \boldsymbol{0}$ for all t, and all slower layers can simply be ignored. The timescale of interest is then the slowest of all of the non-constant layers; hence, no generality is lost by considering only the slowest timescale.

Thus the QSS approximation of the layered system at the slowest timescale is carried out by letting $\mu \to 0$ in (5.13). One possible approach would be to re-partition the reaction set into only two sets, $\mathscr{R}^s = \mathscr{R}^1$ and $\mathscr{R}^f = \bigcup_{l=2}^L \mathscr{R}^l$. The system can then be approximated at the slowest timescale according to the strategy outlined in Section 5.2. Alternatively, the following technique relies on a sequence of successive approximations, starting from approximating the fastest layer L and proceeding up the sequence of timescales until all of the fast layers have been approximated.

Proposition 5.6. *The layered dynamics* (5.13) *at the slowest timescale can be approximated by the*

ODE system

$$\dot{\tilde{\boldsymbol{x}}}^1(t) = \boldsymbol{S}^1\,\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1(t) + \tilde{\boldsymbol{x}}^2(t) + \cdots + \tilde{\boldsymbol{x}}^L(t)\big), \tag{5.14a}$$

$$\tilde{\boldsymbol{x}}^l(t) = \boldsymbol{\phi}^l\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1(t) + \cdots + \tilde{\boldsymbol{x}}^{l-1}(t)\big),\ \textit{for } l = 2,\ldots,L, \tag{5.14b}$$

if the matrix $\boldsymbol{C}^l(\bar{\boldsymbol{v}}^l)'\boldsymbol{U}^l$ *is invertible, for* $l = 2,\ldots,L$.

Here, the static nonlinear functions $\boldsymbol{\phi}^l(\boldsymbol{z})$, *for* $l = 2,\ldots,L$, *satisfy the algebraic equations defining the QSS manifolds* $\boldsymbol{0} = \boldsymbol{S}^l\,\bar{\boldsymbol{v}}^l\big(\boldsymbol{z} + \boldsymbol{\phi}^l(\boldsymbol{z})\big)$, *and the functions* $\bar{\boldsymbol{v}}^l$ *are defined recursively as*

$$\bar{\boldsymbol{v}}^L(\boldsymbol{z}) = \bar{\boldsymbol{v}}(\boldsymbol{z}), \quad \bar{\boldsymbol{v}}^{l-1}(\boldsymbol{z}) = \bar{\boldsymbol{v}}^l\big(\boldsymbol{z} + \boldsymbol{\phi}^l(\boldsymbol{z})\big) \tag{5.15}$$

for decreasing $l = L, (L-1),\ldots,2$.

Proof. See the proof given in Appendix B.1. □

This result implies that a system, layered by flux on multiple timescales, can be approximated by a sequence of L layers. The trajectory of the slowest layer's state $\tilde{\boldsymbol{x}}^1$ drives the state $\tilde{\boldsymbol{x}}^2 = \boldsymbol{\phi}^2(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1)$ of the next slowest timescale. These layers together drive the state $\tilde{\boldsymbol{x}}^3$ of the next slowest layer, and so on down the timescales. The eventual approximated dynamics of the system are equal to the sum $\tilde{\boldsymbol{x}} = \boldsymbol{x}_0 + \sum_{l=1}^{L}\tilde{\boldsymbol{x}}^l$ of the layered states.

5.3.2 Projection form

Note that, when the approximated system is written in the form (5.14), the $L-1$ functions $\boldsymbol{\phi}^l$ must be constructed, which may be relatively impractical. However, similarly to the projection form introduced in Section 5.2.3, the dynamics of the approximate system's state $\tilde{\boldsymbol{x}}$ can be written directly in terms of the faster layers' stoichiometric matrices, as follows.

Definition 5.7 For decreasing $l = L, (L-1),\ldots,2$, define the *interlayer communication matrices* $M^{l,L}$ recursively such that

$$\boldsymbol{M}^{L,L} = -\boldsymbol{U}^L\left(\boldsymbol{C}^L\bar{\boldsymbol{v}}'\boldsymbol{U}^L\right)^{-1}\boldsymbol{C}^L\bar{\boldsymbol{v}}' \tag{5.16a}$$

$$\boldsymbol{M}^{l,L} = -\boldsymbol{U}^l\left(\boldsymbol{C}^l\bar{\boldsymbol{v}}'\boldsymbol{P}^L\boldsymbol{P}^{L-1}\cdots\boldsymbol{P}^{l+1}\boldsymbol{U}^l\right)^{-1}\boldsymbol{C}^l\bar{\boldsymbol{v}}'\boldsymbol{P}^L\boldsymbol{P}^{L-1}\cdots\boldsymbol{P}^{l+1}, \tag{5.16b}$$

where $\boldsymbol{P}^l = \boldsymbol{I} + \boldsymbol{M}^{l,L}$ for $l = 2,3,\ldots,L$. ♢

Proposition 5.8. *Consider the BRN model* $\dot{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x})$ *with initial conditions* $\boldsymbol{x}(0) = \boldsymbol{x}_0$, *and with reactions taking place on L different timescales, admitting the layered decomposition* (5.13). *Define also the non-stiff dynamics*

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{P}^L\boldsymbol{P}^{L-1}\cdots\boldsymbol{P}^2\boldsymbol{S}^1\,\boldsymbol{v}(\tilde{\boldsymbol{x}}), \tag{5.17}$$

where the projections $\boldsymbol{P}^l$ *are defined in* (5.16). *The initial conditions of* (5.17) *are given by* $\boldsymbol{x}_0 + \sum_{l=2}^{L}\tilde{\boldsymbol{x}}^l(\boldsymbol{x}_0)$, *where each* $\boldsymbol{\phi}^l(\boldsymbol{x}_0) = \lim_{t\to\infty}\hat{\boldsymbol{x}}^l(t)$ *is the steady state of layer l, isolated from the slower layers* $k < l$. *For* $\bar{\boldsymbol{v}}^l$ *defined in* (5.15), *the isolated layers that generate the initial conditions have dynamics*

$$\dot{\hat{\boldsymbol{x}}}^l = \boldsymbol{S}^l\,\bar{\boldsymbol{v}}^l(\boldsymbol{x}_0 + \hat{\boldsymbol{x}}^l),$$

initialised from $\hat{\boldsymbol{x}}^l(0) = \boldsymbol{0}$. *As* $\mu \to 0$, *the solution* $\tilde{\boldsymbol{x}}$ *of the dynamics* (5.17) *arbitrarily closely approximates the solution* $\boldsymbol{x}$ *of the original timescale-separated system.*

Proof. See the proof given in Appendix B.2. □

Similarly to the QSS approximation given in (5.14), the expressions (5.16) and (5.17) give an interesting interpretation of how the different timescales interact to produce the overall dynamics at the slowest timescale. The expression (5.17) implies that the dynamics of the isolated slow layer are sequentially projected by $\boldsymbol{P}^2$ through to $\boldsymbol{P}^L$, representing the effect on the slowest timescale of each layer's dynamics at progressively faster timescales. Thus information propagates down the layers from the slowest to the fastest.

However, the definition of each $\boldsymbol{P}^l = \boldsymbol{I} + \boldsymbol{M}^{l,L}$ where $\boldsymbol{M}^{l,L}$ is given by (5.16) implies that the faster layers $k > l$ affect how the slower layers $j < l$ drive layer l. For example, the projection $\boldsymbol{P}^2$ corresponding to the second-slowest layer 2 depends on the projections $\boldsymbol{P}^k$ of the other, faster, layers $k > 2$. Therefore information also propagates up the timescales, so that the way in which an intermediate timescale depends on the slowest dynamics is affected by the presence of faster dynamics.

5.3.3 Example

To illustrate the interaction between layers on $L > 2$ timescales, consider the simple BRN

$$X_1 \xrightleftharpoons[k_{-1}]{k_1} X_2 \xrightleftharpoons[k_{-2}/\mu]{k_2/\mu} X_3 \xrightleftharpoons[k_{-3}/(\mu)^2]{k_3/(\mu)^2} X_4,$$

where all of the $k_{\pm i}$ are assumed to be on the same scale. There are three timescales in this network, corresponding to each reversible reaction. The Jacobian of the (rescaled) flux vector $\bar{\boldsymbol{v}}$ is equal to

$$\bar{\boldsymbol{v}}' = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_{-1} & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_{-2} & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_{-3} \end{bmatrix}.$$

Together with the decompositions $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$ of each layer's stoichiometric matrix, this matrix can be used to construct the projections $\boldsymbol{P}^3$ and $\boldsymbol{P}^2$ according to Definition 5.7. These projections, representing the sequential action of each layer at the slow timescale, are

$$\boldsymbol{P}^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \alpha \\ 0 & 0 & 1-\alpha & 1-\alpha \end{bmatrix}, \quad \boldsymbol{P}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1-\beta & 1-\beta & 1-\beta \\ 0 & \beta & \beta & -(1-\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

for the parameters

$$\alpha = \frac{k_{-3}}{k_3 + k_{-3}}, \quad \beta = \frac{k_2}{k_2 + \alpha k_{-2}}.$$

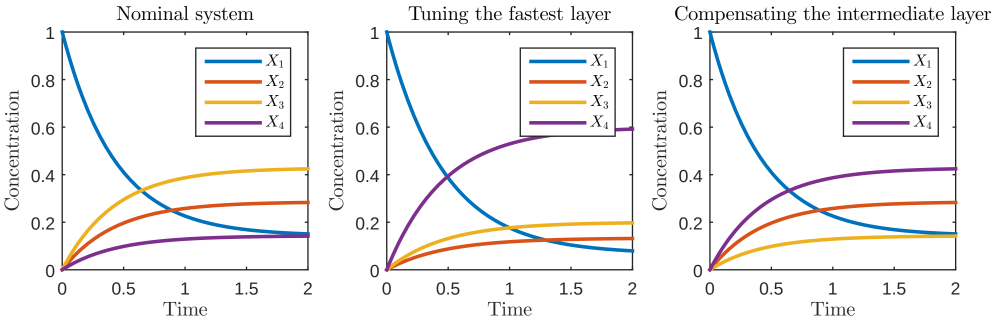


Figure 5.1: Tuning the fastest layer by α. The leftmost plot gives the nominal system, where $k_3 = 1$ and $k_{-3} = 3$ to give $\alpha_1 = 0.75$. The middle plot is the result of reducing this value to $\alpha_2 = 0.25 = 1-\alpha_1$. This parametric perturbation has propagated to affect the dynamics at all timescales. The rightmost plot depicts the system trajectory with the reduced value of $\alpha_2 = 0.25$, but where k_{-2} has been multiplied by a factor of 3. With this compensating action, setting $\alpha_2 = 1-\alpha_1$ inverts the ratio of X_3 to X_4 without affecting the slower timescales.

By Proposition 5.8, the approximated system is then written in projection form on the slowest timescale as

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{P}^3\boldsymbol{P}^2\boldsymbol{S}^1\boldsymbol{v}(\tilde{\boldsymbol{x}}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (1-\beta) & (1-\beta) & (1-\beta) \\ 0 & \alpha\beta & \alpha\beta & \alpha\beta \\ 0 & (1-\alpha)\beta & (1-\alpha)\beta & (1-\alpha)\beta \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_1\tilde{x}_1 \\ k_{-1}\tilde{x}_2 \end{bmatrix}.$$

The slow dynamics are sequentially projected by layer 2 and then layer 3. First, layer 2 instantaneously equilibrates any change in X_2 across X_2 and X_3 in the ratio $1-\beta$ to β. Similarly, the fastest layer equilibrates the resulting change in X_3 across X_3 and X_4, in the ratio α to $1-\alpha$. Taking the product $\boldsymbol{P}^3\boldsymbol{P}^2\boldsymbol{S}^1$ implies that the values $\tilde{\boldsymbol{x}}(t)$ of the approximated state lie in the one-dimensional space spanned by the single basis vector $\left(-1, (1-\beta), \alpha\beta, (1-\alpha)\beta\right)^T$.

The action of each timescale has been identified through the projections $\boldsymbol{P}^3$ and $\boldsymbol{P}^2$. Importantly, the effect $\boldsymbol{P}^2$ of layer 2 depends on the faster layer 3 through the influence of α on β. In particular, it follows from the expression for β that $\alpha < 1$ effectively re-scales the rate k_{-2} of the reverse reaction $X_2 \leftarrow X_3$ to reduce its speed, thereby affecting the impact of the equilibration action of layer 2.

A simple illustration of how information propagates between layers can be demonstrated by tuning the parameter α, as simulated in Figure 5.1. Suppose we intend to increase the concentration of X_4 at the expense of X_3. A simple starting point would be to reduce the value of $\alpha = \frac{k_{-3}}{k_3+k_{-3}}$. If considering the fastest layer in isolation, this implies that the balance of the fastest reactions is shifted to increase X_4 at the expense of X_3. However, the slower layer's parameter β depends on α, such that decreasing α increases β. Therefore, tuning the rates of the fastest reactions also affects the dynamics of all of the slower layers. As illustrated in the middle plot of Figure 5.1, the indirect effect of decreasing α on the slower timescales feeds back to the fastest layer. In the case depicted, where the intention is to reverse the ratio of X_3 to X_4 at each time point, the increase in X_4 is much higher than intended, at the expense of X_1 and X_2.

5.4 Discussion

This chapter has applied the framework of layering by flux, introduced in Chapter 4, to the quasi-steady state approximation of BRNs with timescale-separated reaction rates. The technique is introduced in Section 5.2 in the context of two timescales, and extended in Section 5.3 to systems with dynamics on more than two timescales. Proposition 5.8 produced dynamics in projection form (5.17) for the system under the quasi-steady state approximation. This form identifies the sequential effect of each successively faster layer on the approximated dynamics at the slowest timescale, as the slower layers drive the dynamics of the faster layers. However, the matrices (5.16) that define how each timescale's dynamics responds to the driving force of slower timescales are all also dependent on the presence of faster layers. Therefore information propagates both down the cascade from slow layers to fast, but also up the cascade as the fast layers alter the behaviour of the slow layers. We illustrated this behaviour in Section 5.3.3 with an example of a simple linear system with three timescales, where tuning the reactions on the fastest timescale indirectly affects the behaviour of the intermediate timescale.

This example demonstrates that, when tuning or controlling a multi-scale, layered system to exhibit a certain desired behaviour at a given scale, we must account for the interlayer communication structure causing additional effects at different scales. These may then feed back to the layer being tuned, meaning that the tuned parameters or designed controller is suboptimal. Furthermore, we cannot assume that the cascade of tuning effects is all in one direction: tuning the dynamics of any layer has consequences for the dynamics of all other timescales, including slower ones. The very simple example in Section 5.3.3 is a proof of principle, demonstrating that our framework may enable a design methodology that insulates layers so that control actions affect only specific timescales. For example, the rightmost trajectory in Figure 5.1 shows that, by simultaneously controlling k_{-2} such that the effective reaction rate $k_{-2}\alpha$ is constant, the behaviour of the intermediate scale's layer remains unaffected by tuning α. In this way, a single scale can be controlled by designing appropriate compensating actions that insulate the other timescales from unintended consequences. This example is simple, because the interlayer communication structure is constant. However, in general the projections $\boldsymbol{P}^l(\tilde{\boldsymbol{x}})$ given by Definition 5.7 are also state-dependent. This means that the interlayer communication structure is dynamic; the influence of a fast layer on a slow layer will change over time. Nevertheless, an important direction for potential future work is the formulation of design principles for the reliable parameter-based control of multi-scale systems.

This chapter has considered how to use the layered interconnection structure to analyse the interactions between coupled layers with dynamics that separate in scale. However, there may be cases where the system's reactions or species all occur on the same scale, and the problem of how to choose a layered decomposition remains. The following chapter further develops the concept of the layered interconnection structure in such cases.

Chapter 6

Signal Propagation I: Steady State Analysis

This chapter is concerned with the application of the layered decomposition framework to decompose the steady-state parameter sensitivity analysis of large-scale BRNs, introduced in Section 6.1. Section 6.2 investigates how, given a particular layered decomposition of a BRN, each isolated layer's response to a given parameter perturbation maps to its response when all layers are integrated together. This map can be used to characterise the interlayer communication structure, as shown in Section 6.3. Finally, Section 6.5.1 briefly discusses how the interlayer communication structure of a proposed layered decomposition provides a criterion for measuring its quality, and proposes how this may be used in an algorithm to automatically select the 'best' layered decomposition.

6.1 Parameter Sensitivity Analyses

This section briefly reviews a number of techniques for parameter sensitivity analysis [218]. In general, the goal of parameter sensitivity analysis is to quantify how a particular output of a dynamical system responds to a parameter perturbation. For example, the output in question could be the steady state, which we will use in this chapter. Alternatively, the sensitivity could be measured by the $\mathscr{L}_2$ norm of the change in trajectory, which was introduced in Chapter 2, and which will be revisited in Chapter 7.

Regardless of which output is being measured, its sensitivity to parameter changes may be evaluated in, broadly, two different ways [218]. First, brute-force methods can be used to map how a system's output depends on the values of various parameters as they vary across a broad range. By contrast, local approaches seek to quantify the system's sensitivity to small perturbations around a given set of nominal parameter values. The former has the benefit of being valid across a wide range of possible parameter values, at the expense of computational cost.

6.1.1 Problem statement

This chapter will focus on the technique of quantifying the sensitivity to small perturbations of the vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_r)^T \in \mathbb{R}^r$ of r nominal constant parameter values[1]. The general BRN dynamics (1.2) can be written to explicitly depend on $\boldsymbol{\pi}$, such that

$$\dot{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{x}; \boldsymbol{\pi}), \tag{6.1}$$

with initial conditions $\boldsymbol{x}(0) = \boldsymbol{0}$. Note that the state of (1.2) has been translated to also make the dependence of the dynamics on the initial conditions $\boldsymbol{x}_0$ explicit. The initial conditions $\boldsymbol{x}_0$ can also be thought of as parameters, in the sense that they are a vector of constants that influence the trajectory of the system. Hence, in the translated BRN ODE (6.1) it is explicit that the solution will depend both on the parameters $\boldsymbol{\pi}$ and on the initial conditions $\boldsymbol{x}_0$.

Assume that, for given initial conditions and given parameter values, the BRN with dynamics (6.1) approaches a steady state $\lim_{t\to\infty} \boldsymbol{x}(t) = \bar{\boldsymbol{x}}$. Given $(\boldsymbol{x}_0, \boldsymbol{\pi})$, the trajectory of $\boldsymbol{x}$ from $\boldsymbol{x}(0) = \boldsymbol{0}$ uniquely defines the steady state $\bar{\boldsymbol{x}}(\boldsymbol{x}_0, \boldsymbol{\pi})$ satisfying the equation

$$\boldsymbol{0} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x}_0 + \bar{\boldsymbol{x}};\ \boldsymbol{\pi}). \tag{6.2}$$

It is important to note that (6.2) on its own does not necessarily define $\bar{\boldsymbol{x}}$ uniquely, since there may be multiple solutions; there is an additional constraint that $\bar{\boldsymbol{x}} \in \mathrm{Col}(\boldsymbol{S})$. The goal of local parameter sensitivity analysis is to identify the response of $\bar{\boldsymbol{x}}(\boldsymbol{x}_0, \boldsymbol{\pi})$ to a small change in one or more of the parameters π_i. This response is defined as the derivative $\partial\bar{\boldsymbol{x}}/\partial\pi_i$, if it exists. The sensitivity of the steady state to all parameters can be found as the Jacobian $\partial\bar{\boldsymbol{x}}/\partial\boldsymbol{\pi}$.

Similarly the response of $\bar{\boldsymbol{x}}(\boldsymbol{x}_0, \boldsymbol{\pi})$ to a change in the initial conditions is given by the Jacobian $\partial\bar{\boldsymbol{x}}/\partial\boldsymbol{x}_0$, if it exists. However, for the sake of notation, we will assume in the rest of this chapter that the only possible perturbations will be to $\boldsymbol{\pi}$, and that $\boldsymbol{x}_0$ is fixed and given. In this case, $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}(\boldsymbol{\pi})$ depends on $\boldsymbol{\pi}$ only, and the parameter sensitivity Jacobian is therefore denoted $\mathrm{d}\bar{\boldsymbol{x}}/\mathrm{d}\boldsymbol{\pi}$.

6.1.2 Parameter sensitivity in BRNs

Metabolic Control Analysis (MCA) is a popular technique for parameter sensitivity analysis in the context of BRNs [65]. In this setting, the steady-state response of metabolic networks is subject to perturbations in parameters that represent constant, externally controlled enzyme concentrations. The local analysis introduced in Section 6.1.1 is slightly altered to take place in logarithmic space, by defining sensitivity as $\partial\log(x_i)/\partial\log(\pi_j)$. This approach allows for the derivation of a number of useful summation theorems. Furthermore, a closely related topic that is currently under investigation is parameter identifiability. In particular, systems biology models have been shown in many cases to be ‘sloppy’, where certain measured outputs are insensitive to variations in some parameter values, and extremely fragile to others [81, 208].

Other approaches to parameter sensitivity analysis make use of decomposition techniques. The development of MCA inspired related research in modular [178] and hierarchical [32, 90] control analyses, the aim of which was to apply MCA techniques and results to decomposed

[1] We assume that π_i are constant over all $t \geq 0$, to distinguish them from the more general case of external time-varying inputs to the system, which would be more conventionally denoted u_i.

systems. For example, hierarchical control analysis, as described in Section 4.4, assumes that mass does not flow between subsystems. This special case is used to quantify the interactions between subsystems, allowing for a decomposed approach to parameter sensitivity that relates system responses to the isolated response of individual subsystems. The modular decomposition of parameter sensitivity analysis has also been investigated in the context of BRN model reconstruction [104, 105]. Here, the authors consider a large-scale model consisting of a set of known modules, with an unknown interconnection structure to be determined. By designing a set of experiments on the large-scale system, they show that an 'interaction map' can be experimentally produced that quantifies the interaction between modules. More recent results determine the signs of sensitivities for a given stoichiometry, independently of specific parameter values [189].

The remainder of this chapter will apply the layered framework, introduced in Chapter 4, to decompose parameter sensitivity analysis. Sensitivity will be defined in the form of $\partial \bar{x}_i / \partial \pi_j$, rather than the logarithmic definition used in MCA and its variants. Since sensitivity is quantified in terms of the steady state, we will assume that any layering used to decompose the sensitivity analysis is minimal, so that each layer is also at steady state. The decomposition will result in a relationship between the layers' isolated parameter sensitivity and their responses when integrated in the large-scale system.

6.2 Layered Parameter Sensitivity Analysis

This section will discuss the layered decomposition of parameter sensitivity analysis. The problem statement in Section 6.1.1 defined the steady state $\bar{\boldsymbol{x}}$ of (6.1) as a function of parameters $\boldsymbol{\pi}$ that satisfies the steady state condition (6.2). Before considering a decomposition of this task, we will first outline a technique for the parameter sensitivity analysis of the full-scale system.

In order to illustrate the concepts introduced in this chapter, we use as a running example a model of the set of three reversible reactions

$$X_1 \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} X_2 \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} X_3 \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} X_4, \tag{6.3}$$

where, unlike the example in Section 5.3.3, the reaction rates no longer separate in scale. Assume that the system is layered such that each reversible reaction is in its own layer, whereby

$$\boldsymbol{S} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

If the parameters and initial conditions are set such that $k_{\pm i} = i$ and $\boldsymbol{x}_0 = (1,0,0,0)^T$, the steady state of this system is $\boldsymbol{x}_0 + \bar{\boldsymbol{x}} = (0.25, 0.25, 0.25, 0.25)^T$.

6.2.1 Non-decomposed sensitivity

As described in Chapter 4, the rank r stoichiometric matrix $\boldsymbol{S} = \boldsymbol{UC}$ can be written as the product of the matrices $\boldsymbol{U}$ and $\boldsymbol{C}$, the r columns and rows of which respectively form bases of the column and row space of $\boldsymbol{S}$. Writing $\boldsymbol{x}(t) = \boldsymbol{U\xi}(t)$ for a state $\boldsymbol{\xi}$ taking values $\boldsymbol{\xi}(t) \in \mathbb{R}^r$, the steady state equation (6.2) is equivalent to

$$\boldsymbol{0} = \boldsymbol{Cv}(\boldsymbol{x}_0 + \boldsymbol{U}\bar{\boldsymbol{\xi}};\ \boldsymbol{\pi}) = \boldsymbol{F}(\bar{\boldsymbol{\xi}};\ \boldsymbol{\pi}),$$

where $\bar{\boldsymbol{x}} = \boldsymbol{U}\bar{\boldsymbol{\xi}}$. Note that $\boldsymbol{F}$ also depends on $\boldsymbol{x}_0$, but since it was assumed above that the initial conditions will not be perturbed, this is not made explicit.

As in the proof of Corollary 5.4, the Implicit Function Theorem (IFT) can be applied to the level set of $\boldsymbol{F} = \boldsymbol{0}$ in order to determine how $\bar{\boldsymbol{\xi}}$, and therefore $\bar{\boldsymbol{x}}$, depends on $\boldsymbol{\pi}$. The IFT requires the matrix

$$\frac{\partial \boldsymbol{F}}{\partial \bar{\boldsymbol{\xi}}}(\bar{\boldsymbol{\xi}};\boldsymbol{\pi}) = \boldsymbol{Cv_x}(\boldsymbol{x}_0 + \boldsymbol{U}\bar{\boldsymbol{\xi}};\boldsymbol{\pi})\boldsymbol{U}$$

to be evaluated, where $\boldsymbol{v_x}$ denotes the Jacobian of $\boldsymbol{v}(\boldsymbol{x};\boldsymbol{\pi})$ with respect to the first argument, $\boldsymbol{x}$. If this matrix, evaluated at the steady state, is invertible then there exists an open set Π containing the nominal parameter vector $\boldsymbol{\pi}$, and a vector-valued function $\boldsymbol{\phi}$ such that the steady state $\bar{\boldsymbol{\xi}} = \boldsymbol{\phi}(\tilde{\boldsymbol{\pi}})$ varies continuously with $\tilde{\boldsymbol{\pi}} \in \Pi$. The key result for parameter sensitivity analysis is that the Jacobian of $\boldsymbol{\phi}$ at $\boldsymbol{\pi}$ is

$$\frac{\mathrm{d}\bar{\boldsymbol{\xi}}}{\mathrm{d}\boldsymbol{\pi}} = -\big(\boldsymbol{Cv_x}(\boldsymbol{x}_0 + \boldsymbol{U}\bar{\boldsymbol{\xi}};\boldsymbol{\pi})\boldsymbol{U}\big)^{-1}\boldsymbol{Cv_\pi}(\boldsymbol{x}_0 + \boldsymbol{U}\bar{\boldsymbol{\xi}};\boldsymbol{\pi}),$$

where $\boldsymbol{v_\pi}$ is the Jacobian of $\boldsymbol{v}(\boldsymbol{x};\boldsymbol{\pi})$ with respect to the components of $\boldsymbol{\pi}$. Since $\bar{\boldsymbol{x}} = \boldsymbol{U}\bar{\boldsymbol{\xi}}$, this derivative is pre-multiplied by $\boldsymbol{U}$ to give

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}}{\mathrm{d}\boldsymbol{\pi}} = -\boldsymbol{U}\big(\boldsymbol{Cv_x}(\boldsymbol{x}_0 + \bar{\boldsymbol{x}};\boldsymbol{\pi})\boldsymbol{U}\big)^{-1}\boldsymbol{Cv_\pi}(\boldsymbol{x}_0 + \bar{\boldsymbol{x}};\boldsymbol{\pi}), \tag{6.4}$$

quantifying the sensitivity of $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}(\boldsymbol{\pi})$ to the parameters in $\boldsymbol{\pi}$.

Continued For the example system (6.3) with a fixed initial condition vector $\boldsymbol{x}_0$, the parameter vector $\boldsymbol{\pi} = (k_1, k_{-1}, k_2, k_{-2}, k_3, k_{-3})^T$ determines the steady state $\bar{\boldsymbol{x}}$. For the chosen value of $\boldsymbol{\pi} = (1,1,2,2,3,3)^T$ the steady state is given by $\boldsymbol{x}_0 + \bar{\boldsymbol{x}} = (0.25, 0.25, 0.25, 0.25)$. In order to characterise the sensitivity of this steady state to perturbations in $\boldsymbol{\pi}$, consider the Jacobian of the flux vector $\boldsymbol{v}(\boldsymbol{x};\boldsymbol{\pi})$. Each of the partial Jacobians is given by

$$\boldsymbol{v_x}(\boldsymbol{x};\boldsymbol{\pi}) = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_{-1} & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_{-2} & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_{-3} \end{bmatrix}, \quad \boldsymbol{v_\pi}(\boldsymbol{x};\boldsymbol{\pi}) = \begin{bmatrix} x_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & x_4 \end{bmatrix}, \tag{6.5}$$

while a feasible pair of column and row space matrices satisfying $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{C}$ are

$$\boldsymbol{U} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Substituting these matrices into (6.4) and performing the 3 × 3 matrix inverse gives a sensitivity value of

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}}{\mathrm{d}\boldsymbol{\pi}} = \begin{bmatrix} -0.1875 & 0.1875 & -0.0625 & 0.0625 & -0.0208 & 0.0208 \\ 0.0625 & -0.0625 & -0.0625 & 0.0625 & -0.0208 & 0.0208 \\ 0.0625 & -0.0625 & 0.0625 & -0.0625 & -0.0208 & 0.0208 \\ 0.0625 & -0.0625 & 0.0625 & -0.0625 & 0.0625 & -0.0625 \end{bmatrix}.$$

Interestingly, the sensitivity of state x_4 has equal magnitude (in two different directions) to perturbations of any parameter, while state x_1 is far less sensitive to perturbations in $k_{\pm 3}$ than it is to perturbations in $k_{\pm 1}$. Even state x_3, which takes part in reactions with parameters $k_{\pm 3}$, is minimally sensitive to those parameters. This initial analysis suggests that a layered approach would give more insight. ◇

In general, the formula (6.4) for parameter sensitivity requires the inverse of a $r \times r$ matrix to be calculated, where r is the dimension of the system. If this is large, then quantifying a system's sensitivity in a single step becomes difficult. As with many large-scale systems analysis tasks, decomposition is a natural simplification strategy. By decomposing the steady-state analysis of a system, the response of individual subsystems to parameter perturbations can be identified. This can then be mapped to the response of the integrated system by tracking how these effects propagate across the interlayer connection structure.

6.2.2 Direct and indirect parameter sensitivity

The dynamics (6.1) can now be decomposed into L layers by finding $\boldsymbol{S}^l$, for $l = 1, \dots, L$, such that $\boldsymbol{S} = \boldsymbol{S}^1 + \cdots + \boldsymbol{S}^L$. Each layer's state $\boldsymbol{x}^l$ takes values $\boldsymbol{x}^l(t) \in \mathbb{R}^N$ with dynamics

$$\dot{\boldsymbol{x}}^l = \boldsymbol{S}^l \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{x}^1 + \cdots + \boldsymbol{x}^L;\ \boldsymbol{\pi}), \tag{6.6}$$

and with initial conditions $\boldsymbol{x}^l(0) = \boldsymbol{0}$. It is assumed that the layering is minimal, which implies that the layered dynamics inherit the stability of the original system. Hence, there exist $\bar{\boldsymbol{x}}^l$ satisfying

$$\boldsymbol{0} = \boldsymbol{S}^l \boldsymbol{v}(\boldsymbol{x}_0 + \bar{\boldsymbol{x}}^1 + \cdots + \bar{\boldsymbol{x}}^L;\ \boldsymbol{\pi}), \tag{6.7}$$

with $\boldsymbol{x}^l(t) \to \bar{\boldsymbol{x}}^l$ as $t \to \infty$ for each $l = 1, \dots, L$. As shown in the derivation of (4.8), for each layer l, the input from the other layers $k \neq l$ can be aggregated into a single input $\boldsymbol{u}^l = \sum_{k \neq l} \boldsymbol{x}^k$, with a resulting steady state input of $\bar{\boldsymbol{u}}^l = \sum_{k \neq l} \bar{\boldsymbol{x}}^k$. Furthermore, using the decomposition of $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$ to write the layered dynamics in a minimal realisation (4.13) implies that the steady state condition (6.7) is equivalent to

$$\boldsymbol{0} = \boldsymbol{C}^l \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l + \bar{\boldsymbol{u}}^l;\ \boldsymbol{\pi}) =: \boldsymbol{F}^l(\bar{\boldsymbol{\xi}}^l;\ \bar{\boldsymbol{u}}^l, \boldsymbol{\pi}), \tag{6.8}$$

where each layer's minimally realised state variable $\boldsymbol{\xi}^l$ taking values $\boldsymbol{\xi}^l \in \mathbb{R}^{r^l}$ has steady state such that $\bar{\boldsymbol{x}}^l = \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l$.

As in the non-decomposed case of Section 6.2.1, the direct response of the steady state $\bar{\boldsymbol{x}}^l$ of layer l to perturbations in $\boldsymbol{\pi}$ can be derived using the Implicit Function Theorem. Assuming that the square matrix $\boldsymbol{C}^l \boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0 + \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l + \bar{\boldsymbol{u}}^l;\ \boldsymbol{\pi})\boldsymbol{U}^l$ is invertible, this theorem implies that the sensitivity of the steady state $\bar{\boldsymbol{x}}^l(\bar{\boldsymbol{u}}^l, \boldsymbol{\pi})$ to perturbations in $\boldsymbol{\pi}$ is

$$\frac{\partial \bar{\boldsymbol{x}}^l}{\partial \boldsymbol{\pi}} = -\boldsymbol{U}^l \big(\boldsymbol{C}^l \boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0 + \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l + \bar{\boldsymbol{u}}^l;\ \boldsymbol{\pi})\boldsymbol{U}^l\big)^{-1} \boldsymbol{C}^l \boldsymbol{v}_{\boldsymbol{\pi}}(\boldsymbol{x}_0 + \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l + \bar{\boldsymbol{u}}^l;\ \boldsymbol{\pi}). \tag{6.9}$$

Here the matrices $\boldsymbol{v}_{\boldsymbol{x}}$ and $\boldsymbol{v}_{\boldsymbol{\pi}}$ again correspond to the Jacobians of the flux vector $\boldsymbol{v}(\boldsymbol{x};\ \boldsymbol{\pi})$ as a function of $\boldsymbol{x}$ and $\boldsymbol{\pi}$ respectively.

Continued Consider the layered decomposition of the system (6.3) of three reversible reactions. The expressions for $\boldsymbol{v}_{\boldsymbol{x}}$ and $\boldsymbol{v}_{\boldsymbol{\pi}}$ given in (6.5) remain valid, but now we consider the matrices

$$\boldsymbol{U}^1 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{C}^1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

decomposing $\boldsymbol{S}^1 = \boldsymbol{U}^1 \boldsymbol{C}^1$. Substituting these values into (6.9) gives the isolated response of layer 1 as

$$\frac{\partial \bar{\boldsymbol{x}}^1}{\partial \boldsymbol{\pi}} = \begin{bmatrix} -0.1250 & 0.1250 & 0 & 0 & 0 & 0 \\ 0.1250 & -0.1250 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Similarly, the other two layers have isolated responses of

$$\frac{\partial \bar{\boldsymbol{x}}^2}{\partial \boldsymbol{\pi}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0625 & 0.0625 & 0 & 0 \\ 0 & 0 & 0.0625 & -0.0625 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial \bar{\boldsymbol{x}}^3}{\partial \boldsymbol{\pi}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0417 & 0.0417 \\ 0 & 0 & 0 & 0 & 0.0417 & -0.0417 \end{bmatrix}.$$

Clearly, perturbations to $k_{\pm l}$ only have direct effects in layer l, for each $l = 1, 2, 3$. However, on summing these matrices we find that the response of $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}^1 + \bar{\boldsymbol{x}}^2 + \bar{\boldsymbol{x}}^3$ does not match the result found earlier. ♢

The running example illustrates the general fact that the expression (6.9) will not fully capture the response of $\bar{\boldsymbol{x}}^l(\bar{\boldsymbol{u}}^l, \boldsymbol{\pi})$ to perturbations in $\boldsymbol{\pi}$. This is because the direct response of the layer propagates as an internal perturbation to the other layers' steady states $\bar{\boldsymbol{x}}^k$ for $k \neq l$. This means that, in many cases indirectly, the other layers' steady states $\bar{\boldsymbol{x}}^k$ are also dependent on the component of $\boldsymbol{\pi}$ directly affecting the steady state of layer l. Thus, perturbations to $\boldsymbol{\pi}$ will also cause the input steady state $\bar{\boldsymbol{u}}^l = \sum_{j \neq l} \bar{\boldsymbol{x}}^j$ to respond. This is made explicit by rewriting (6.8) as

$$\boldsymbol{0} = \boldsymbol{C}^l \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^l \bar{\boldsymbol{\xi}}^l + \bar{\boldsymbol{u}}^l;\ \boldsymbol{\pi}) = \boldsymbol{F}^l\big(\bar{\boldsymbol{\xi}}^l;\ \bar{\boldsymbol{u}}^l(\boldsymbol{\pi}), \boldsymbol{\pi}\big), \tag{6.10}$$

where it is now explicit that $\bar{\boldsymbol{u}}^l$ also depends on $\boldsymbol{\pi}$.

Proposition 6.1. *The sensitivity of the steady state* $\bar{\boldsymbol{x}}^l\big(\bar{\boldsymbol{u}}^l(\boldsymbol{\pi}),\boldsymbol{\pi}\big)$ *of layer* l *to perturbations in the parameter vector* $\boldsymbol{\pi}$ *is*

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}^l}{\mathrm{d}\boldsymbol{\pi}} = -\boldsymbol{U}^l\big(\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0+\bar{\boldsymbol{x}};\ \boldsymbol{\pi})\boldsymbol{U}^l\big)^{-1}\Big[\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{\pi}}(\boldsymbol{x}_0+\bar{\boldsymbol{x}};\ \boldsymbol{\pi})+\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0+\bar{\boldsymbol{x}};\ \boldsymbol{\pi})\frac{\mathrm{d}\bar{\boldsymbol{u}}^l}{\mathrm{d}\boldsymbol{\pi}}\Big], \tag{6.11}$$

for the system steady state $\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}^l + \bar{\boldsymbol{u}}^l$.

Proof. Again apply the Implicit Function Theorem to $\boldsymbol{F}^l$ in (6.10), to find that

$$\frac{\mathrm{d}\bar{\boldsymbol{\xi}}^l}{\mathrm{d}\boldsymbol{\pi}} = -\left(\frac{\partial \boldsymbol{F}^l}{\partial\bar{\boldsymbol{\xi}}^l}\right)^{-1}\left(\frac{\partial \boldsymbol{F}^l}{\partial\boldsymbol{\pi}}+\frac{\partial \boldsymbol{F}^l}{\partial\bar{\boldsymbol{u}}^l}\frac{\mathrm{d}\bar{\boldsymbol{u}}^l}{\mathrm{d}\boldsymbol{\pi}}\right).$$

By three simple applications of the chain rule, the required derivatives of $\boldsymbol{F}^l$ can be found. Since $\bar{\boldsymbol{x}}^l = \boldsymbol{U}^l\bar{\boldsymbol{\xi}}^l$, we premultiply this derivative by $\boldsymbol{U}^l$ to give the result. □

Thus for each layer l there are two types of sensitivity to a parameter change. The expression (6.9) for $\partial\bar{\boldsymbol{x}}^l/\partial\boldsymbol{\pi}$ will be termed the *isolated response*. This quantity represents the sensitivity of layer l to perturbations in $\boldsymbol{\pi}$, assuming that the steady states of the other layers making up $\bar{\boldsymbol{u}}^l$ are held constant. This response is often intuitively obvious. For example, if layering by flux then the vector of reaction rates will be partitioned across the layers. Hence, varying one of the reaction rates k_i will have a direct effect on only one layer l, and the other direct sensitivities $\partial\bar{\boldsymbol{x}}^j/\partial k_i$ will be zero for $j \neq l$. Of course, in an integrated system the response of layers $j \neq l$ to perturbations in k_i may not be zero; this is where the *integrated response* (6.11) applies.

The expression (6.11) for $\mathrm{d}\bar{\boldsymbol{x}}^l/\mathrm{d}\boldsymbol{\pi}$ considers the case where layer l is integrated with all of the other layers of the system. It includes the integrated responses $\mathrm{d}\bar{\boldsymbol{x}}^k/\mathrm{d}\boldsymbol{\pi}$ of each of the other layers $k \neq l$ as additional perturbations to layer l. These additional perturbations capture the effect of that layer's context on its response to parameter perturbations. The resulting responses can often be counter-intuitive, since layers are likely to respond to perturbations of parameters not directly involved with any of its reactions. We will show later in this chapter that we can construct a detailed characterisation of the layered architecture by tracking how direct responses propagate between layers.

6.2.3 Mapping isolated to integrated responses

The integrated response $\mathrm{d}\bar{\boldsymbol{x}}^l/\mathrm{d}\boldsymbol{\pi}$ given by (6.11) can be reformulated by writing the input $\bar{\boldsymbol{u}}^l = \sum_{j\neq l}\bar{\boldsymbol{x}}^j$ as the aggregated contribution of all of the other layers. In this case,

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}^l}{\mathrm{d}\boldsymbol{\pi}} = \frac{\partial\bar{\boldsymbol{x}}^l}{\partial\boldsymbol{\pi}} - \sum_{k\neq l}\left[\boldsymbol{U}^l\left(\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}\boldsymbol{U}^l\right)^{-1}\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}\frac{\mathrm{d}\bar{\boldsymbol{x}}^k}{\mathrm{d}\boldsymbol{\pi}}\right] = \frac{\partial\bar{\boldsymbol{x}}^l}{\partial\boldsymbol{\pi}} + \sum_{k\neq l}\left[\boldsymbol{M}^l(\bar{\boldsymbol{x}})\frac{\mathrm{d}\bar{\boldsymbol{x}}^k}{\mathrm{d}\boldsymbol{\pi}}\right]$$

for each $l = 1,\ldots,L$, where $\partial\bar{\boldsymbol{x}}^l/\partial\boldsymbol{\pi}$ is given by (6.9) and where, similarly to (5.5), the interlayer communication matrix

$$\boldsymbol{M}^l(\bar{\boldsymbol{x}}) = -\boldsymbol{U}^l\big(\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0+\bar{\boldsymbol{x}};\ \boldsymbol{\pi})\boldsymbol{U}^l\big)^{-1}\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}(\boldsymbol{x}_0+\bar{\boldsymbol{x}};\ \boldsymbol{\pi}) \tag{6.12}$$

can be defined for each $l = 1, \dots, L$. Substituting this expression for $\boldsymbol{M}^l$ into each instance of (6.11) gives the matrix equation

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{M}^1 & \dots & -\boldsymbol{M}^1 \\ -\boldsymbol{M}^2 & \boldsymbol{I} & \dots & -\boldsymbol{M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ -\boldsymbol{M}^L & -\boldsymbol{M}^L & \dots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \mathrm{d}\bar{\boldsymbol{x}}^1/\mathrm{d}\boldsymbol{\pi} \\ \mathrm{d}\bar{\boldsymbol{x}}^2/\mathrm{d}\boldsymbol{\pi} \\ \vdots \\ \mathrm{d}\bar{\boldsymbol{x}}^L/\mathrm{d}\boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \partial\bar{\boldsymbol{x}}^1/\partial\boldsymbol{\pi} \\ \partial\bar{\boldsymbol{x}}^2/\partial\boldsymbol{\pi} \\ \vdots \\ \partial\bar{\boldsymbol{x}}^L/\partial\boldsymbol{\pi} \end{bmatrix}.$$

Assuming that this matrix is invertible, the map from the isolated responses to the integrated responses is given by

$$\begin{bmatrix} \mathrm{d}\bar{\boldsymbol{x}}^1/\mathrm{d}\boldsymbol{\pi} \\ \mathrm{d}\bar{\boldsymbol{x}}^2/\mathrm{d}\boldsymbol{\pi} \\ \vdots \\ \mathrm{d}\bar{\boldsymbol{x}}^L/\mathrm{d}\boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{M}^1 & \dots & -\boldsymbol{M}^1 \\ -\boldsymbol{M}^2 & \boldsymbol{I} & \dots & -\boldsymbol{M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ -\boldsymbol{M}^L & -\boldsymbol{M}^L & \dots & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \partial\bar{\boldsymbol{x}}^1/\partial\boldsymbol{\pi} \\ \partial\bar{\boldsymbol{x}}^2/\partial\boldsymbol{\pi} \\ \vdots \\ \partial\bar{\boldsymbol{x}}^L/\partial\boldsymbol{\pi} \end{bmatrix}. \tag{6.13}$$

This equation means that the parameter sensitivity analysis can be decomposed. First, each layer's isolated sensitivity $\partial\bar{\boldsymbol{x}}^l/\partial\boldsymbol{\pi}$ to a parameter perturbation can be evaluated. This is then mapped according to (6.13) to find each layer's sensitivity $\mathrm{d}\bar{\boldsymbol{x}}^l/\mathrm{d}\boldsymbol{\pi}$ when integrated with the other layers in the system.

The expression in (6.13) is closely related to a similar relationship between local and global steady-state sensitivities in the framework of modular response analysis. For example, in [104], the authors 'experimentally' perturb a system and infer a matrix similar to that given in (6.13), to describe the map from the theoretical direct response to the observed global response. A similar matrix is found in [32] for steady states in block-diagonal stoichiometric matrices, and in [190] at a number of timepoints to infer *dynamic* interconnection architecture. In all of those cases, the matrix mapping local to global responses is not necessarily structured. In comparison, the matrix in (6.13) is highly structured; this is a consequence of the layered approach ensuring that each response $\partial\bar{\boldsymbol{x}}^l/\partial\boldsymbol{\pi}$ is calculated directly in $\mathbb{R}^N$. In the rest of this chapter, we will illustrate the use of this matrix structure for gaining a network-level understanding of how signals propagate between layers.

Continued Recall the derivation in Section 6.2.2 of the isolated response $\partial\bar{\boldsymbol{x}}/\partial\boldsymbol{\pi}$ of each layer of the system (6.3) to perturbations in $\boldsymbol{\pi}$. Substituting the matrices $\boldsymbol{U}^l$, $\boldsymbol{C}^l$, and $\boldsymbol{v}_{\boldsymbol{x}}$ into (6.12) gives three interlayer communication matrices

$$\boldsymbol{M}^1 = \begin{bmatrix} -0.5 & 0.5 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{M}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.5 & 0 \\ 0 & 0.5 & -0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{M}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0.5 & -0.5 \end{bmatrix}$$

that define the map (6.13). The resulting integrated responses of each layer to perturbations in

$\boldsymbol{\pi}$ are given by

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}^1}{\mathrm{d}\boldsymbol{\pi}} = \begin{bmatrix} -0.1875 & 0.1875 & -0.0625 & 0.0625 & -0.0208 & 0.0208 \\ 0.1875 & -0.1875 & 0.0625 & -0.0625 & 0.0208 & -0.0208 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}^2}{\mathrm{d}\boldsymbol{\pi}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.125 & 0.125 & -0.125 & 0.125 & -0.0417 & 0.0417 \\ 0.125 & -0.125 & 0.125 & -0.125 & 0.0417 & -0.0417 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\frac{\mathrm{d}\bar{\boldsymbol{x}}^3}{\mathrm{d}\boldsymbol{\pi}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0625 & 0.0625 & -0.0625 & 0.0625 & -0.0625 & 0.0625 \\ 0.0625 & -0.0625 & 0.0625 & -0.0625 & 0.0625 & -0.0625 \end{bmatrix},$$

which sum to give the overall system response $\mathrm{d}\bar{\boldsymbol{x}}/\mathrm{d}\boldsymbol{\pi}$ found in Section 6.2.1. $\diamondsuit$

In Section 6.2.1, the layered decomposition of parameter sensitivity analysis was motivated by the inversion of an $r \times r$ matrix $\boldsymbol{C}\boldsymbol{v}'\boldsymbol{U}$ in (6.4), which we assume to be large. However, the matrix inverse used in (6.13) is an $LN \times LN$ matrix, which is in fact larger. We will show in the following section that this map can be reformulated as an $r \times r$ matrix (assuming a minimal layering) but, nevertheless, this is still a large matrix inverse problem on the same order as that in the non-decomposed case. However, the key property of the inverse in (6.13) is that the matrix we need to invert is clearly very highly structured in comparison to $\boldsymbol{C}\boldsymbol{v}'\boldsymbol{U}$, which may be useful in speeding the required computation. The following section explores this structure in more detail.

6.3 Uncovering Layered Architectures

Define the matrix $\boldsymbol{P}$ and the vectors $\boldsymbol{R}$ and $\boldsymbol{r}$ as

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{0} & \boldsymbol{M}^1 & \dots & \boldsymbol{M}^1 \\ \boldsymbol{M}^2 & \mathbf{0} & \dots & \boldsymbol{M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}^L & \boldsymbol{M}^L & \dots & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \mathrm{d}\bar{\boldsymbol{x}}^1/\mathrm{d}\boldsymbol{\pi} \\ \mathrm{d}\bar{\boldsymbol{x}}^2/\mathrm{d}\boldsymbol{\pi} \\ \vdots \\ \mathrm{d}\bar{\boldsymbol{x}}^L/\mathrm{d}\boldsymbol{\pi} \end{bmatrix}, \quad \boldsymbol{r} = \begin{bmatrix} \partial\bar{\boldsymbol{x}}^1/\partial\boldsymbol{\pi} \\ \partial\bar{\boldsymbol{x}}^2/\partial\boldsymbol{\pi} \\ \vdots \\ \partial\bar{\boldsymbol{x}}^L/\partial\boldsymbol{\pi} \end{bmatrix},$$

so that the map (6.13) from the isolated layer responses $\boldsymbol{r}$ to the integrated layer responses $\boldsymbol{R}$ can be written

$$\boldsymbol{R} = (\boldsymbol{I} - \boldsymbol{P})^{-1}\boldsymbol{r}. \tag{6.14}$$

This section will investigate the structure of $\boldsymbol{P}$ in more detail, and how this can be used to characterise the interconnection structure of the decomposed system.

6.3.1 The inter-layer graph

The $N \times N$ block of $\boldsymbol{P}$ above in position (l,k) is $\boldsymbol{M}^l$, as defined by (6.12). Similarly to the final point of Corollary 5.4, this matrix is independent of the particular bases $\boldsymbol{U}^l$ and $\boldsymbol{C}^l$ used in the decomposition $\boldsymbol{S}^l = \boldsymbol{U}^l\boldsymbol{C}^l$. Therefore, without any loss of generality, assume for the remainder of this chapter that the matrix $\boldsymbol{U}^l$ used to construct $\boldsymbol{M}^l$, the columns of which form a basis for $\mathrm{Col}(\boldsymbol{S}^l)$, is equal to $\boldsymbol{U}^l$ as defined in the following lemma.

Lemma 6.2. *For a given layered decomposition* (6.6)*, each* $\boldsymbol{M}^l$ *has a compact singular value decomposition*

$$\boldsymbol{M}^l = -\boldsymbol{U}^l\boldsymbol{\Sigma}^l\boldsymbol{V}^l,$$

where $\boldsymbol{\Sigma}^l$ *is an* $r^l \times r^l$ *diagonal matrix of singular values* $\sigma_i^l \geq 1$ *defined by*

$$\boldsymbol{\Sigma}^l = (\boldsymbol{V}^l\boldsymbol{U}^l)^{-1}$$

for $r^l = \mathrm{rank}(\boldsymbol{S}^l)$ *denoting the dimension of layer* l*.*

Proof. Since $(\boldsymbol{M}^l)^2 = -\boldsymbol{M}^l$, it follows that $\boldsymbol{U}^l\boldsymbol{\Sigma}^l\boldsymbol{V}^l\boldsymbol{U}^l\boldsymbol{\Sigma}^l\boldsymbol{V}^l = \boldsymbol{U}^l\boldsymbol{\Sigma}^l\boldsymbol{V}^l$. Since each of $\boldsymbol{U}^l$, $\boldsymbol{V}^l$ and $\boldsymbol{\Sigma}^l$ are full rank, this implies that $\boldsymbol{\Sigma}^l(\boldsymbol{V}^l\boldsymbol{U}^l) = (\boldsymbol{V}^l\boldsymbol{U}^l)\boldsymbol{\Sigma}^l = \boldsymbol{I}$. Thus $(\boldsymbol{V}^l\boldsymbol{U}^l)^{-1}$ is a diagonal matrix of singular values. Since each row $\boldsymbol{v}_i^l$ of $\boldsymbol{V}^l$ and column $\boldsymbol{u}_i^l$ of $\boldsymbol{U}^l$ is a vector of length n with norm 1, the singular value is given by $\sigma_i^l = 1/(\boldsymbol{v}_i^l\boldsymbol{u}_i^l) \geq 1$. □

The following definition uses the SVD decomposition of each of the L matrices $\boldsymbol{M}^l$ to construct a weighted, directed, L-partite graph on a set $\mathcal{V}$ of $|\mathcal{V}| = r = r^1 + \cdots + r^L$ nodes. This graph is intended to represent the means by which perturbations to the steady state of each layer can propagate to other layers. The map $\boldsymbol{R} = (\boldsymbol{I}-\boldsymbol{P})^{-1}\boldsymbol{r}$ given above can be shown to be closely related to this graphical representation of the interlayer communication structure.

Definition 6.3 Define the *inter-layer graph* on $r^1 + \cdots + r^L = r$ nodes as a weighted, directed, L-partite graph with the block adjacency matrix $\boldsymbol{A}$ given by

$$\boldsymbol{A}_{lk} = \begin{cases} -\left(\boldsymbol{V}^l\boldsymbol{U}^l\right)^{-1}\boldsymbol{V}^l\boldsymbol{U}^k \in \mathbb{R}^{r^l \times r^k} & \text{for } l \neq k, \\ \boldsymbol{0} \in \mathbb{R}^{r^l \times r^k} & \text{else,} \end{cases}$$

for $l,k = 1,\ldots,L$, where $\left(\boldsymbol{V}^l\boldsymbol{U}^l\right)^{-1}$ is a diagonal $r^l \times r^l$ matrix of scalars $\sigma_1^l \geq \cdots \geq \sigma_{r^l}^l \geq 1$. ◊

The node set $\mathcal{V}$ of the inter-layer graph can be naturally partitioned into L subsets $\mathcal{V}^l$ of $|\mathcal{V}^l| = r^l$ nodes each. Each subset $\mathcal{V}^l$ corresponds to a layer l, while each node $i \in \mathcal{V}^l$ represents a column $\boldsymbol{u}_{\tau^l(i)}^l$ of $\boldsymbol{U}^l$, where the index $\tau^l(\cdot)$ defines a bijective mapping between $\mathcal{V}^l$ and the integers $\{1,\ldots,r^l\}$ indexing the columns of $\boldsymbol{U}^l$. The only edges in this graph are between nodes $(i,j) \in \mathcal{V}^l \times \mathcal{V}^k$ for $l \neq k$, making this an L-partite graph. All of the edge weights from nodes in $\mathcal{V}^k$ to nodes in $\mathcal{V}^l$ are in the block $\boldsymbol{A}_{lk}$; the weight $(\boldsymbol{A}_{lk})_{ij}$ of the edge (i,j) represents the gain from a perturbation of $\bar{\boldsymbol{x}}^k$ in the unit direction $\boldsymbol{u}_{\tau^k(j)}^k$ causing a subsequent perturbation of $\bar{\boldsymbol{x}}^l$ in the unit direction $\boldsymbol{u}_{\tau^l(i)}^l$. This weight may be negative, since the downstream perturbation in layer l may be anti-parallel to $\boldsymbol{u}_{\tau^l(i)}^l$.

Before investigating some properties of this graph, the following proposition demonstrates that it is closely related to the matrix $\boldsymbol{P}$ in (6.14).

Proposition 6.4. *Define the block diagonal matrix* $\boldsymbol{Q} = \operatorname{diag}(\boldsymbol{U}_1, \ldots, \boldsymbol{U}_L)$. *The matrices* $\boldsymbol{A}$ *and* $\boldsymbol{P}$ *satisfy*

$$\boldsymbol{Q}\boldsymbol{A}^n = \boldsymbol{P}^n\boldsymbol{Q}, \quad \textit{and} \quad \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{A})^{-1} = (\boldsymbol{I} - \boldsymbol{P})^{-1}\boldsymbol{Q},$$

for all $n \in \mathbb{N}$.

Proof. First, we show that $\boldsymbol{QA} = \boldsymbol{PQ}$. For $l \neq k$, the (l,k)-th block of $\boldsymbol{PQ}$ is equal to

$$\boldsymbol{M}^l\boldsymbol{U}^k = -\boldsymbol{U}^l\left(\boldsymbol{V}^l\boldsymbol{U}^l\right)^{-1}\boldsymbol{V}^l\boldsymbol{U}^k = \boldsymbol{U}^l\boldsymbol{A}_{lk},$$

which is clearly equal to the (l,k)-th block of $\boldsymbol{QA}$. Since each of the (l,l)-th blocks along the diagonal are zero, it follows that $\boldsymbol{QA} = \boldsymbol{PQ}$.

Suppose $\boldsymbol{QA}^n = \boldsymbol{P}^n\boldsymbol{Q}$. It follows that $\boldsymbol{QA}^{n+1} = \boldsymbol{QA}^n\boldsymbol{A} = \boldsymbol{P}^n\boldsymbol{QA} = \boldsymbol{P}^n\boldsymbol{PQ} = \boldsymbol{P}^{n+1}\boldsymbol{Q}$, and by induction the first result holds for all $n \in \mathbb{N}$. Finally, it follows from $\boldsymbol{QA} = \boldsymbol{PQ}$ that $\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{A}) = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Q}$. By appropriately pre- and post-multiplying each side of this equality, the second result follows. □

Now consider the vector $\boldsymbol{r}$ in (6.14). By the formula (6.9) given for each $\partial\bar{\boldsymbol{x}}/\partial\boldsymbol{\pi}$, this vector can be re-written as $\boldsymbol{r} = \boldsymbol{Qr}'$. The lth vector component of $\boldsymbol{r}' = (\boldsymbol{r}_1', \ldots, \boldsymbol{r}_L')^T$ is equal to the vector $\boldsymbol{r}_l' = -\left(\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{x}}\boldsymbol{U}^l\right)^{-1}\boldsymbol{C}^l\boldsymbol{v}_{\boldsymbol{\pi}}$. Each element $(\boldsymbol{r}_l')_i$ of this vector represents the size of the direct response to a perturbation in $\boldsymbol{\pi}$ of layer l in the unit direction $\boldsymbol{u}_i^l$. Therefore Proposition 6.4 implies that

$$\boldsymbol{R} = \left(\boldsymbol{I} - \boldsymbol{P}\right)^{-1}\boldsymbol{Qr}' = \boldsymbol{Q}\left(\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{r}', \tag{6.15}$$

and hence the map from isolated to integrated responses can be identified with the adjacency matrix $\boldsymbol{A}$ of the inter-layer graph.

After deriving the map from isolated to integrated responses in the form (6.15), it still remains to exploit the structure of this matrix to simplify its inversion. The first approach to this is to consider expanding (6.15) into the matrix series

$$\boldsymbol{R} = \boldsymbol{Q}\left(\boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \ldots\right)\boldsymbol{r}',$$

assuming that this sum converges. Each power $\boldsymbol{A}^n$ of the adjacency matrix corresponds to the n-hop adjacency matrix. Thus the (l,k)th block $\boldsymbol{A}_{lk}^{(n)}$ of $\boldsymbol{A}^n$ gives the gain from the local response $\boldsymbol{r}_k'$ of layer k to its contribution to the global response of layer l, taken only along all paths of length n. Hence the total gain from the local response $\boldsymbol{r}_k'$ in layer k to the global response $\boldsymbol{R}_l$ of layer l is given by the the sum $\sum_n \boldsymbol{A}_{lk}^{(n)}$ of all of the n-hop gains.

Continued The working example (6.3) is a simple illustration of this concept, since each layer is of rank one, and therefore corresponds to a single node in a network of $L = 3$ nodes. For the local responses $\boldsymbol{r}$ found above, the matrix $\boldsymbol{r}'$ satisfying $\boldsymbol{r} = \boldsymbol{Qr}'$ is given by

$$\boldsymbol{r}' = \begin{bmatrix} 0.1768 & -0.1768 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0884 & -0.0884 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0589 & -0.0589 \end{bmatrix},$$

where $\boldsymbol{Q} = \frac{1}{\sqrt{2}}\operatorname{diag}(\boldsymbol{U}^1, \boldsymbol{U}^2, \boldsymbol{U}^3)$ for the column matrices $\boldsymbol{U}^l$ used in Section 6.2.2. More importantly, the interlayer adjacency matrices (which are in fact scalars, since each layer has only one associated node) define the interlayer graph adjacency

$$A = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix}.$$

The matrix inverse required in the map $\boldsymbol{R} = \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{r}'$ is equal to

$$(\boldsymbol{I} - \boldsymbol{A})^{-1} = \begin{bmatrix} 1.5 & 1 & 0.5 \\ 1 & 2 & 1 \\ 0.5 & 1 & 1.5 \end{bmatrix}.$$

The first column of this inverse matrix tells us that in the integrated system, a direct perturbation to layer 1 affects the output directions of all of the layers. The magnitude of the direct perturbation to layer 1 is multiplied by 1.5 to give the integrated response of layer 1. The magnitude of the direct response is not scaled at all in the response of layer 2, and is attenuated in its effect on layer 3. A similar argument can be made for how the isolated responses of layer 2 and layer 3 are attenuated or amplified when integrated into the system.

The matrix inverse can be related to the adjacency $\boldsymbol{A}$ of the interlayer graph through the series $(\boldsymbol{I} - \boldsymbol{A})^{-1} = \boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \ldots$. Taking each term of this series in turn, we can conclude that a direct perturbation of magnitude ϵ to layer 1 propagates in one step to perturb layer 2 in its output direction, with magnitude $\epsilon/2$. At the second step, this perturbation to layer 2 propagates both back to layer 1 and to layer 3, now with magnitude $\epsilon/4$ each. Each of these propagate back to layer 2 in the third stage, where the magnitude of the response is the sum $\epsilon/8 + \epsilon/8 = \epsilon/4$. The signal attenuation continues around the network as the steps continue. ◊

In the example above, it is clear that layer 1 and layer 3 do not directly communicate; perturbations to layer 1 reach layer 3 after two propagations, having been scaled by a factor of 1/4. In general, the interlayer graph encodes the information about how signals propagate through a layered network. In the case where layers have rank $r^l > 1$, the interlayer adjacencies are matrices. The following section discusses two important properties of these matrices that can capture some of the important features of how steady-state responses to parameter perturbations propagate across a layered network in this more complicated case.

6.3.2 Properties of the inter-layer graph

6.3.2.1 Gain

Consider first the $r^l \times r^k$ real-valued matrix $\boldsymbol{A}_{lk} = -(\boldsymbol{V}^l\boldsymbol{U}^l)^{-1}\boldsymbol{V}^l\boldsymbol{U}^k$. This matrix defines how perturbations in each of the r^k directions of layer k propagate to directly perturb each of the r^l directions in layer l. Thus, if $\boldsymbol{A}_{lk} = \boldsymbol{0}$ (or, equivalently, $\boldsymbol{V}^l\boldsymbol{U}^k = \boldsymbol{0}$) then the steady state of layer l does not directly respond to perturbations in the steady state of layer k. We can write the p, q-th element of the inter-layer adjacency explicitly as

$$(\boldsymbol{A}_{lk})_{pq} = \sigma_p^l \boldsymbol{v}_p^l \boldsymbol{u}_q^k$$

for $p = 1,\dots,r^l$ and $q = 1,\dots,r^k$, for the rows $\boldsymbol{v}_p^l$ of $\boldsymbol{V}^l$ and columns $\boldsymbol{u}_q^k$ of $\boldsymbol{U}^k$. Thus the p,q-th element of $\boldsymbol{A}_{lk}$ is σ_p^l times the cosine of the angle between the p-th row of the input space of layer i and the q-th column of the output space of layer j.

Suppose that the steady state of layer k is perturbed in such a way that its response $\Delta\bar{\boldsymbol{x}}^k$ is solely in the q-th direction $\boldsymbol{u}_q^k$ of $\boldsymbol{U}^k$, with magnitude ϵ. This perturbation propagates to layer l such that the resulting response in $\bar{\boldsymbol{x}}^l$ is a multiple of the q-th column of $\boldsymbol{A}_{lk}$, given by

$$\Delta\bar{\boldsymbol{x}}^l = \epsilon\left[\sigma_1^l(\boldsymbol{v}_1^l\boldsymbol{u}_q^k) \quad \dots \quad \sigma_{r^l}^l(\boldsymbol{v}_{r^l}^l\boldsymbol{u}_q^k)\right]^T.$$

Therefore, the gain as the layer k response propagates into layer l is equal to

$$\left(\sum_{p=1}^{r^l}\left(\sigma_p^l(\boldsymbol{v}_p^l\boldsymbol{u}_q^k)\right)^2\right)^{1/2}.$$

By repeating this argument for a general response $\Delta\bar{\boldsymbol{x}}^k \in \mathrm{Col}(\boldsymbol{U}^k)$ of layer k to a perturbation, the norm of the resulting direct response $\Delta\bar{\boldsymbol{x}}^l$ in layer l is given by the expression

$$\left\|\Delta\bar{\boldsymbol{x}}^l\right\| = \left\|\boldsymbol{A}_{lk}\Delta\bar{\boldsymbol{x}}^k\right\| = \left(\sum_{p=1}^{r^l}\left(\sigma_p^l(\boldsymbol{v}_p^l\Delta\bar{\boldsymbol{x}}^k)\right)^2\right)^{1/2}.$$

Since each σ_p^l is bounded above and below such that $1 \le \sigma_p^l \le \sigma_1^l$, this satisfies the upper and lower bound

$$\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\Delta\bar{\boldsymbol{x}}^k\right)^2\right)^{1/2} \le \left\|\boldsymbol{A}_{lk}\Delta\bar{\boldsymbol{x}}^k\right\| \le \sigma_1^i\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\Delta\bar{\boldsymbol{x}}^k\right)^2\right)^{1/2}.$$

These inequalities imply that the gain in response magnitude from $\Delta\bar{\boldsymbol{x}}^k$ to $\Delta\bar{\boldsymbol{x}}^l$ can be bounded above and below. The bracketed terms

$$\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\Delta\bar{\boldsymbol{x}}^k\right)^2\right)^{1/2}$$

are equal to the length of the layer k response $\Delta\bar{\boldsymbol{x}}^k$ projected into in the subspace spanned by the rows of $\boldsymbol{V}^l$. Hence, if $\Delta\bar{\boldsymbol{x}}^k$ is orthogonal to the row space of $\boldsymbol{V}^l$, this gain is zero and the perturbation does not propagate.

The formula above can be used to find the induced gain $\|\boldsymbol{A}_{lk}\|$ by taking the supremum over $\mathscr{U}^k = \{\boldsymbol{u} \in \mathrm{Col}(\boldsymbol{U}^k) \mid \|\boldsymbol{u}\| = 1\}$, since

$$\sup_{\boldsymbol{u}\in\mathscr{U}^k}\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\boldsymbol{u}\right)^2\right)^{1/2} \le \|\boldsymbol{A}_{lk}\| \le \sigma_1^l \sup_{\boldsymbol{u}\in\mathscr{U}^k}\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\boldsymbol{u}\right)^2\right)^{1/2}. \tag{6.16}$$

This means that, if there exists $\boldsymbol{u} \in \mathrm{Col}(\boldsymbol{U}^k) \cap \mathrm{Row}(\boldsymbol{V}^l)$, then $\sum(\boldsymbol{v}_p^l\boldsymbol{u})^2 = 1$ for this $\boldsymbol{u}$. Therefore the induced gain $\|\boldsymbol{A}_{lk}\| \ge 1$ amplifies (or, at least, does not attenuate) a space of possible perturbations $\Delta\bar{\boldsymbol{x}}^k$ as they propagate from layer k to layer l. Similarly, if $\left(\sum_{p=1}^{r^l}\left(\boldsymbol{v}_p^l\boldsymbol{u}\right)^2\right)^{1/2} \le 1/\sigma_1^l$ for all unit vectors $\boldsymbol{u} \in \mathscr{U}^k$, then all steady state responses propagating from layer k to layer l must be attenuated. Indeed, if all $\boldsymbol{u} \in \mathrm{Col}(\boldsymbol{U}^k)$ are orthogonal to the row space $\mathrm{Row}(\boldsymbol{V}^l)$, then no response can propagate from layer k to layer l at all.

Note that the expression in (6.16) for the maximum length of the projection of $\boldsymbol{u} \in \mathscr{U}^k$ into $\mathrm{Row}(\boldsymbol{V}^l)$ can be used to define the angle between the vector spaces $\mathrm{Col}(\boldsymbol{U}^k)$ and $\mathrm{Row}(\boldsymbol{V}^l)$ as

$$\cos\left(\angle\left[\mathrm{Col}(\boldsymbol{U}^k), \mathrm{Row}(\boldsymbol{V}^l)\right]\right) = \sup_{\boldsymbol{u}\in\mathscr{U}^k} \left(\sum_{p=1}^{r^l} \left(\boldsymbol{v}_p^l \boldsymbol{u}\right)^2\right)^{1/2} \leq 1.$$

Then, the conditions on whether the block $\boldsymbol{A}_{lk}$ attenuates or amplifies responses as they propagate from layer k to layer l can be expressed in terms of the orthogonality between subspaces by substituting this angle definition into (6.16).

6.3.2.2 Rank

As well as determining the gain of each block $\boldsymbol{A}_{lk}$, the rank of each interlayer adjacency matrix can also be found. The following proposition relates the rank of $\boldsymbol{A}_{lk}$ to the spaces $\mathrm{Row}(\boldsymbol{V}^l)$ and $\mathrm{Col}(\boldsymbol{U}^k)$.

Proposition 6.5. *The inter-layer adjacency matrix $\boldsymbol{A}_{lk}$ has rank*

$$\mathrm{rank}(\boldsymbol{A}_{lk}) \leq \min\left[r^l - \dim\left(\mathrm{Row}(\boldsymbol{V}^l) \cap \mathrm{Col}(\boldsymbol{U}^k)^\perp\right), r^k - \dim\left(\mathrm{Row}(\boldsymbol{V}^l)^\perp \cap \mathrm{Col}(\boldsymbol{U}^k)\right)\right],$$

where r^k is reduced by the number of linearly independent vectors in $\mathrm{Col}(\boldsymbol{U}^k)$ *that are perpendicular to* $\mathrm{Row}(\boldsymbol{V}^l)$*, and vice versa for the reduction in r^l.*

Proof. Construct a matrix $\boldsymbol{W}$ consisting of w columns that form an orthonormal basis of the intersection subspace $\mathrm{Row}(\boldsymbol{V}^l)^\perp \cap \mathrm{Col}(\boldsymbol{U}^k)$. This basis can be extended by an additional $r^k - w$ linearly independent vectors to a basis for $\mathrm{Col}(\boldsymbol{U}^k)$. The additional vectors form the columns of the matrix $\boldsymbol{X}$, so that the columns of $\tilde{\boldsymbol{U}}^k = [\boldsymbol{W}, \boldsymbol{X}]$ form an orthonormal basis for the entire space $\mathrm{Col}(\boldsymbol{U}^k)$. There is an invertible $r^k \times r^k$ change of basis matrix $\boldsymbol{T}^k$ so that $\boldsymbol{U}^k = \tilde{\boldsymbol{U}}^k \boldsymbol{T}^k$.

Now $\mathrm{rank}(\boldsymbol{A}_{lk}) = \mathrm{rank}(\boldsymbol{V}^l\boldsymbol{U}^k) = \mathrm{rank}(\boldsymbol{V}^l\tilde{\boldsymbol{U}}^k\boldsymbol{T}^k) = \mathrm{rank}(\boldsymbol{V}^l\tilde{\boldsymbol{U}}^k)$. By construction, the product $\boldsymbol{V}^l\tilde{\boldsymbol{U}}^k$ is equal to the matrix

$$\boldsymbol{V}^l\tilde{\boldsymbol{U}}^k = \begin{bmatrix}\boldsymbol{V}^l\boldsymbol{W} & \boldsymbol{V}^l\boldsymbol{X}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{V}^l\boldsymbol{X}\end{bmatrix},$$

consisting of $r^k - w$ non-zero columns of length r^l. Thus, the rank of this product is bounded above by $r^k - w$ for $w = \dim\left(\mathrm{Row}(\boldsymbol{V}^l)^\perp \cap \mathrm{Col}(\boldsymbol{U}^k)\right)$.

To find the other upper bound, note that $\mathrm{rank}(\boldsymbol{A}_{lk}^T) = \mathrm{rank}(\boldsymbol{A}_{lk})$. A similar argument to that made above, swapping $\boldsymbol{V}^l$ and $\boldsymbol{U}^k$ for $(\boldsymbol{U}^k)^T$ and $(\boldsymbol{V}^l)^T$ respectively, produces the required upper bound. □

6.3.2.3 Summaries of interlayer graphs

The results above relate the gain and rank of $\boldsymbol{A}_{lk}$ to the orthogonality of the spaces $\mathrm{Row}(\boldsymbol{V}^l)$ and $\mathrm{Col}(\boldsymbol{U}^k)$. Upper and lower estimates of the gain $\|\boldsymbol{A}_{lk}\|$ may determine whether the magnitude of perturbations to a given layer k are attenuated or amplified when they propagate to layer l. The rank, or more precisely $\dim\left(\mathrm{Row}(\boldsymbol{V}^l)^\perp \cap \mathrm{Col}(\boldsymbol{U}^k)\right)$, quantifies the number of independent response directions of layer k that do not have any effect on layer l. Similarly, $\dim\big(\mathrm{Row}(\boldsymbol{V}^l) \cap \mathrm{Col}(\boldsymbol{U}^k)^\perp\big)$ quantifies the number of independent response directions of layer l that cannot be affected by layer k.

These quantities can be summarised in two directed graphs on L nodes, where each node now corresponds to a layer. The first of these is the interlayer gain graph, with adjacency $\boldsymbol{A}_{gain} \in \mathbb{R}^{L\times L}$ defined by

$$(\boldsymbol{A}_{gain})_{lk} = \begin{cases} \sigma_1^l \cos\Big(\angle\big(\mathrm{Row}(\boldsymbol{V}^l), \mathrm{Col}(\boldsymbol{U}^k)\big)\Big) & \text{for } l \neq k, \\ 0 & \text{else.} \end{cases}$$

Then if $(\boldsymbol{A}_{gain})_{lk} < 1$, the direct propagation of a perturbation from layer k to layer l is guaranteed to be attenuated. Furthermore, the nature of the induced gain means that we can bound the gain of the l,kth block of the n-hop adjacency matrix $\boldsymbol{A}^n$ by

$$\|\boldsymbol{A}_{lk}^{(n)}\| \leq (\boldsymbol{A}_{gain}^n)_{lk},$$

thereby finding an upper bound on the gain as a perturbation propagates from layer k to layer l along all possible paths of length n. Of course, this upper bound is likely to be extremely conservative, but gives an indication of how quickly the series $\boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \ldots$ converges to $(\boldsymbol{I} - \boldsymbol{A})^{-1}$.

Similarly, we can define the interlayer rank graph, with adjacency $\boldsymbol{A}_{rank} \in \mathbb{R}^{L\times L}$ given by

$$(\boldsymbol{A}_{rank})_{lk} = \begin{cases} \mathrm{rank}(\boldsymbol{V}^l\boldsymbol{U}^k) & \text{for } l \neq k, \\ 0 & \text{else.} \end{cases}$$

The integer-valued edge weights of this matrix correspond to the number of independent output directions of layer k that can drive a response in layer l. Alternatively, the weight represents the number of independent response directions of layer l that can be driven by perturbations to layer k.

6.4 Example: Goldbeter–Koshland Switches

Consider the following model of a cascade of three Goldbeter–Koshland switches, which has been applied to the sensitivity analysis of MAPK signalling cascades [26, 76]:

$$X + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C_1 \xrightarrow{m_1} X^\star + S, \qquad X^\star + E_2 \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} C_2 \xrightarrow{m_2} X + E_2, \tag{6.17a}$$

$$Y + X^\star \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} C_3 \xrightarrow{m_3} Y^\star + X^\star, \qquad Y^\star + E_4 \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} C_4 \xrightarrow{m_4} Y + E_4, \tag{6.17b}$$

$$Z + Y^\star \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} C_5 \xrightarrow{m_5} Z^\star + Y^\star, \qquad Z^\star + E_6 \underset{k_{-6}}{\overset{k_6}{\rightleftharpoons}} C_6 \xrightarrow{m_6} Z + E_6. \tag{6.17c}$$

Here, S denotes an enzyme that acts as a signal modulating the forward rate of the first switch $X \rightleftharpoons X^\star$. Each of E_2, E_4, and E_6 is an enzyme that modulates the reverse reaction rate of each switch. The notation $\cdot^\star$ denotes the activated form of the protein X, Y, or Z. The key to the cascade is that the activated protein $X^\star$ modulates the forward rate of $Y \rightleftharpoons Y^\star$, and similarly $Y^\star$ modulates the forward rate of $Z \rightleftharpoons Z^\star$. Note that a modular decomposition of this network according to the rows of (6.17) will demonstrate retroactivity; while $X^\star$ is an input to the $Y/Y^\star$ switch, the mechanism of interconnection means that the dynamics of the $X/X^\star$ switch will change.

The model of this cascade has stoichiometric matrix $\boldsymbol{S}$ such that $\operatorname{rank}(\boldsymbol{S}) = 9$. We assume that the state vector is given by $\boldsymbol{x} = (x, x^\star, y, y^\star, z, z^\star, c_1, c_2, c_3, c_4, c_5, c_6)^T$. The enzyme concentrations e_i can be neglected from this model because of the conservation relations $c_i(t) + e_i(t) = E_{i,tot}$ for the constants $E_{i,tot}$, for $i = 2, 4, 6$.

Assume that $k_i = 2$, $k_{-i} = 1$, and $m_i = 3$ for each $i = 1, \ldots, 6$, and that the initial conditions are such that $x(0) = y(0) = z(0) = 1$ and all other concentrations 0. Furthermore, let $S = E_2 = E_4 = E_6 = 0.1$. Under these assumptions, the steady state of this network is equal to

$$\begin{aligned}
1+\bar{x} &= 0.4669, & \bar{x}^\star &= 0.4669, & \bar{c}_1 &= 0.0189, & \bar{c}_2 &= 0.0189,\\
1+\bar{y} &= 0.1215, & \bar{y}^\star &= 0.7917, & \bar{c}_3 &= 0.0284, & \bar{c}_4 &= 0.0284,\\
1+\bar{z} &= 0.0762, & \bar{z}^\star &= 0.8635, & \bar{c}_5 &= 0.0302, & \bar{c}_6 &= 0.0302.
\end{aligned}$$

However, if S is then set to $S = 5$, the steady state of the system moves to

$$\begin{aligned}
1+\bar{x} &= 0.0125, & \bar{x}^\star &= 0.8960, & \bar{c}_1 &= 0.0309, & \bar{c}_2 &= 0.0309,\\
1+\bar{y} &= 0.0663, & \bar{y}^\star &= 0.8441, & \bar{c}_3 &= 0.0297, & \bar{c}_4 &= 0.0297,\\
1+\bar{z} &= 0.0717, & \bar{z}^\star &= 0.8678, & \bar{c}_5 &= 0.0303, & \bar{c}_6 &= 0.0303.
\end{aligned}$$

In both cases, the high concentration of $\bar{z}^\star$ implies that the system is switched on, even when the signal $S = 0.1$ is relatively low. The effect of the cascade is to amplify the signal S. We now consider small perturbations around the two given values of $S = 0.1$ and $S = 5$, and observe the effect on the steady state in each case.

The vector $\partial \boldsymbol{v}/\partial S$, representing the Jacobian of $\boldsymbol{v}(\boldsymbol{x}_0 + \bar{\boldsymbol{x}}; \boldsymbol{\pi})$ with respect to the single parameter S, has value $k_1(x(0) + \bar{x})$ in the first component, with the other 17 components equal to zero. This results in the two cases for the parameter sensitivity of

$$\begin{aligned}
\frac{\mathrm{d}\bar{\boldsymbol{x}}}{\mathrm{d}S} &= \begin{bmatrix} -2.98 & 2.78 & -0.66 & 0.62 & -0.06 & 0.05 & 0.09 & 0.09 & 0.02 & 0.02 & 0.001 & 0.001 \end{bmatrix}^T,\\
\frac{\mathrm{d}\bar{\boldsymbol{x}}}{\mathrm{d}S} &= \begin{bmatrix} -0.003 & 0.002 & -0.0002 & 0.0002 & 0 & 0 & 0.0001 & 0.0001 & 0 & 0 & 0 & 0 \end{bmatrix}^T,
\end{aligned}$$

for $S = 0.1$ and $S = 5$ respectively. Clearly the steady state $\bar{z}^\star$ of 'on' is extremely robust for both parameter values, whereas the 'on' state $\bar{x}^\star$ of the first switch is extremely sensitive to perturbations around the parameter $S = 0.1$. Thus the cascade robustifies the switch to perturbations in the input signal.

A layered analysis of this system now allows us to consider how the structure of this system acts to robustify the output signal. Let the reaction set $\mathscr{R} = \{R_1, \ldots, R_{18}\}$ be indexed according to the rows of (6.17). It can be shown that layering by flux, using the partition of $\mathscr{R}$ into $L = 3$ disjoint subsets $\mathscr{R}^1 = \{R_1, \ldots, R_6\}$, $\mathscr{R}^2 = \{R_7, \ldots, R_{12}\}$, and $\mathscr{R}^3 = \{R_{13}, \ldots, R_{18}\}$, produces a minimal layering where $r^l = \operatorname{rank}(\boldsymbol{S}^l) = 3$ for $l = 1, 2, 3$. Each layer corresponds to a row of (6.17).

By partitioning the reactions into $\mathscr{R}^l$, the network's parameters have also been partitioned. Thus the perturbation to S will only directly affect the steady state of layer 1. The formula (6.9) implies that the isolated response of layer 1 is

$$\frac{\partial \bar{\boldsymbol{x}}^1}{\partial S} = \begin{bmatrix} -2.97 & 2.79 & 0 & 0 & 0 & 0 & 0.09 & 0.09 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$
$$\frac{\partial \bar{\boldsymbol{x}}^1}{\partial S} = \begin{bmatrix} -0.0025 & 0.0024 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

for $S = 0.1$ and $S = 5$ respectively, with $\partial \bar{\boldsymbol{x}}^2 / \partial S = \partial \bar{\boldsymbol{x}}^3 / \partial S = 0$ for both cases. We focus on the former case, where the steady state of layer 1 is relatively sensitive to perturbations around $S = 0.1$.

The SVD decomposition of each $\boldsymbol{M}^l = -\boldsymbol{U}^l (\boldsymbol{V}^l \boldsymbol{U}^l)^{-1} \boldsymbol{V}^l$ results in an interlayer graph adjacency $\boldsymbol{A}$ defining a tripartite network on 9 nodes, in 3 groups of 3. The summary matrices

$$\boldsymbol{A}_{gain} = \begin{bmatrix} 0 & 0.6345 & 0 \\ 0.4393 & 0 & 0.1481 \\ 0 & 0.1597 & 0 \end{bmatrix}, \quad \boldsymbol{A}_{rank} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

imply that, similarly to the one-dimensional layers in the worked example earlier in this chapter, layer 1 and layer 3 do not directly influence one another, but do so through layer 2. The ranks imply that for each of the communicating pairs layer k and layer l, two of the possible response directions of layer k cannot propagate to layer l. Equivalently, two of the possible response directions of layer l cannot be affected by layer k.

In particular, the steady state of layer 3 can only respond to *any* perturbation of layer 1 or layer 2 in a direction parallel to

$$\begin{bmatrix} 0 & 0 & 0 & -0.0168 & -0.7234 & 0.6898 & 0 & 0 & 0 & 0 & 0.0168 & 0.0168 \end{bmatrix}^T.$$

The fourth and sixth components of this response vector show that an increase in signal of $\bar{z}^\star$ is proportional to a decrease in the contribution $\bar{y}^{\star 2}$ of layer 2 to $\bar{y}^\star$, thereby further demonstrating the use of layered analysis in quantifying retroactivity.

The angle between the possible outputs of layer 2 and the input space of layer 3 is such that $\cos\Big(\angle\big(\mathrm{Row}(\boldsymbol{V}^3), \mathrm{Col}(\boldsymbol{U}^2)\big)\Big) = 0.0808$. Therefore, only a small component of the response of layer 2 can propagate to layer 3, and hence the steady state $\bar{z}^\star$ remains relatively insulated from many of the perturbations to either layer 1 or layer 2; in particular, to perturbations in the signal S.

6.5 Discussion

This chapter has described how a layered decomposition of a network's parameter sensitivity analysis can help describe the layered architecture of the network. In particular, by comparing the steady-state responses of isolated layers to parameter perturbations with their integrated responses, one can construct a graph that reflects the communication structure amongst the layers. This final section identifies two important potential developments to the analysis of steady-state signal propagation across a layered network.

6.5.1 Algorithmic layer detection

Throughout this chapter it has been assumed that a layered decomposition is given. In Section 6.3 we have shown that the ranks and gains of the interlayer adjacencies $\boldsymbol{A}_{ij}$ are important quantities that can be used to characterise how layers communicate, through the filtering and attenuation of each layer's response as it propagates across an integrated network. However, suppose that multiple possible layered decompositions of a large-scale BRN are proposed, or even all possible decompositions. The analysis techniques introduced in this chapter may provide a quantification of how well-partitioned the system's response is by each candidate layered decomposition. This can then be used as criterion in an algorithmic approach to detecting layered structures within a large-scale BRN.

This chapter has distinguished between an interconnection from layer k to layer l being weak, which we defined as a small gain $\|\boldsymbol{A}_{lk}\|$, and sparse, which was defined as a relatively small value of $\operatorname{rank}(\boldsymbol{A}_{lk})$. We propose that one criterion for a 'good' decomposition is one where the inter-layer communication structure is as sparse as possible, measured by the rank-deficiency in the inter-layer communication matrices. Such a rank deficiency in $\boldsymbol{A}_{lk}$ allows us to conclude that only a small number of possible outputs from layer k can influence layer l (alternatively, that layer k can only drive a small number of responses in layer l), and hence that the layering reflects an underlying structure to the steady-state response of the network.

Similarly to the techniques used to evaluate prospective decompositions of networks into communities [145], a measure of the quality of a given layered decomposition is required. Given the low-rank interlayer signal propagation criterion, one reasonable strategy is to minimise each $r^{lk} = \operatorname{rank}(\boldsymbol{A}_{lk})$ in comparison to its upper bound $r^{lk} \leq \min(r^l, r^k)$. We assume that the given layering is not trivial, meaning that all layers have positive dimension $r^l > 0$. This allows us to define the single quantity

$$Q_{rank} = \sum_{l \neq k} \frac{r^{lk}}{\min(r^l, r^k)},$$

which summarises the total rank deficiency of each $\boldsymbol{A}^{lk}$ compared to its maximum possible value. Here, Q is the 1-norm of a matrix $\boldsymbol{B}$ with off-diagonal elements $b_{lk} = r^{lk} / \min(r^l, r^k)$. Minimising the 1-norm of a vector is a heuristic approach to optimising its sparsity [161].

Another heuristic approach to rank minimisation of any matrix $\boldsymbol{X}$ is to minimise its nuclear norm, which is defined as the sum of the singular values of $\boldsymbol{X}$ [161, 162]. Let the singular values of $\boldsymbol{V}^l \boldsymbol{U}^k$ be denoted σ_p^{lk} for $p = 1, \ldots, \min(r^l, r^k)$, ordered so that $\sigma_{p+1}^{lk} \leq \sigma_p^{lk}$. To ensure that the atomic norms $\sum_p \sigma_p^{lk}$ are minimised relative to the ranks of the layer l and layer k, we again divide by $\min(r^l, r^k)$ to define the quantity

$$Q_{atomic} = \sum_{l \neq k} \frac{1}{\min(r^l, r^k)} \left(\sum_{p=1}^{\min(r^l, r^k)} \sigma_p^{lk} \right)$$

which is to be minimised. This heuristic should also ensure that the largest singular value σ_1^{lk} of $\boldsymbol{V}^l \boldsymbol{U}^k$ is relatively small. Thus, the inequality (6.16) implies that minimising Q_{atomic} also gives preference to layered decompositions with weak, low-rank interconnections.

For example, consider two different minimal layered decompositions of the glycolysis example introduced in Section 4.5.2. The first of these decompositions was found at the end of that section, and produces the interlayer communication structure such that

$$\boldsymbol{A}_{rank,1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 3 \\ 2 & 2 & 0 \end{bmatrix}, \; Q_{rank,1} = 2.333, \; Q_{atomic,1} = 1.7668,$$

where the subscript 1 denotes the first adjacency. Consider another possible decomposition of this BRN into three layers, given by partitioning the reaction set into disjoint subsets $\mathscr{R}^1 = \{R_{16}, R_{17}\}$, $\mathscr{R}^2 = \{R_{15}, R_{18}, R_{19}, R_{20}, R_{21}, R_{23}, R_{24}\}$, and $\mathscr{R}^3 = \mathscr{R} \setminus (\mathscr{R}^1 \cup \mathscr{R}^2)$. The minimal layering defined according to the strategy presented in Section 4.4 has an interconnection structure such that

$$\boldsymbol{A}_{rank,2} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix}, \; Q_{rank,2} = 2.333, \; Q_{atomic,2} = 1.6146,$$

where layer 1 of dimension $r^1 = 2$ is downstream of layer 2 and layer 3. These two layers communicate with one another such that only 4 of the $r^2 = 6$ possible output directions of layer 2 are driven by layer 3, and 4 out of $r^3 = 12$ possible response directions of layer 3 are driven by layer 2. Here, although Q_{rank} does not distinguish between these two layerings, the quantity Q_{atomic} would imply that the second layering is preferable.

More work would be required in order to develop a fully automated layered decomposition algorithm. Minimising one of the two quantities Q_{rank} and Q_{atomic} are both heuristic strategies to ensuring low-rank interlayer connections. As with modularity, these numbers do not mean anything on their own; they only make sense when comparing two layered decompositions of the same network. Another connection with community detection is the difficulty of the task of actually algorithmically selecting candidate decompositions to compare. Many community detection methods are heuristic approaches to finding a decomposition of the nodes that is close to optimal. There is no similar result here, and so brute-force methods, perhaps with a greedy sequential approach of decomposing one layer into two a total of $L-1$ times, are currently the best strategy for detecting a layered decomposition.

6.5.2 Dynamic layered architecture

Like Chapter 5, this chapter has dealt with the question of how to partition the reactions to form a layered decomposition. However, by attempting to measure the quality of a decomposition by how well it partitions a steady-state analysis, we lose a lot of the information contained in the dynamic response. The system described in Section 4.5.1 provides an example of how this loss of information can give misleading results. Consider the layering defined by the stoichiometric matrices $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ as given in (4.18). By the quantification of the interlayer communication structure given in this chapter, we find that $\boldsymbol{A}^{12} = \boldsymbol{A}^{21} = \mathbf{0}$.

The steady state analysis described in this chapter would imply that the layers are therefore independent. However, this is only true at steady state: the dynamics of the two layers are not independent, since the dynamics of layer 2 depend on the trajectory of layer 1. Thus we need to proceed beyond a steady state sensitivity analysis to fully characterise the interconnection structure of a system decomposed into a given set of layers.

Consider again the layered dynamics (4.7), in minimal realisation, where each state $\boldsymbol{x}^l$ is written $\boldsymbol{x}^l = \boldsymbol{U}^l \boldsymbol{\xi}^l$ so that

$$\dot{\boldsymbol{\xi}}^l(t) = \boldsymbol{C}^l \boldsymbol{v}\left(\boldsymbol{x}_0 + \boldsymbol{U}^k \boldsymbol{\xi}^k(t) + \sum_{j \neq k} \boldsymbol{U}^j \boldsymbol{\xi}^j(t)\right).$$

Here the effect of layer k is separated from that of the other layers. If the right-hand side of this expression is independent of $\boldsymbol{\xi}^k$, then layer k cannot directly influence the dynamics of layer l. The criterion for independence is that the Jacobian of the right-hand side of these dynamics with respect to $\boldsymbol{\xi}^k$ is identically zero. That is, if

$$\boldsymbol{C}^l \boldsymbol{v}'(\boldsymbol{x}_0 + \boldsymbol{x}) \boldsymbol{U}^k = \boldsymbol{0} \tag{6.18}$$

for all $\boldsymbol{x} \in \mathrm{Col}(\boldsymbol{S})$, then layer l does not directly depend on layer k. Here, $\boldsymbol{v}'(\boldsymbol{x})$ denotes the Jacobian of the flux vector $\boldsymbol{v}$ with respect to $\boldsymbol{x}$.

The condition (6.18) is clearly stronger than the steady state definition of independence, which is only required to hold for $\boldsymbol{x} = \bar{\boldsymbol{x}}$. Future developments of the analysis of layered interconnection structure need to account for the interlayer communication matrices $\boldsymbol{M}^l$ and $\boldsymbol{A}^{lk}$ varying with the trajectory of $\boldsymbol{x}(t)$. As a first step towards this goal, the following chapter will relate the perturbation of one layer's trajectory, measured as $\|\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k\|^2$, to the resulting perturbation $\|\boldsymbol{x}^l - \tilde{\boldsymbol{x}}^l\|^2$ of a downstream layer's trajectory.

Chapter 7

Signal Propagation II: Cascaded Dynamics

A key limitation of the SOS programming technique for dissipativity analysis described in Chapter 2 is its scalability when applied to large-dimensional systems. This provides the motivation for this chapter, which applies the layering framework discussed in this thesis to decompose the task of constructing a storage function. The contribution of this chapter is to provide a framework of *Structured Storage Functions*, which are formulated to reflect the layered architecture of the system.

In Chapter 3, dissipativity analysis was applied to quantify the difference between the outputs of a nominal model and of a structurally or parametrically perturbed model. By applying decomposition to this problem, these perturbations can be localised to subsystems. Similarly to the steady-state analysis of the previous chapter, the response of the rest of the model to such localised perturbations can then be quantified. Hence we show how decomposition, in the specific case of systems that can be decomposed into a cascade of subsystems, can be used to make the estimation of the dynamic responses to parametric and structural perturbations more practically feasible.

7.1 Dissipativity and Structure

Recall the definition of dissipativity in Chapter 2 for a nonlinear dynamical system (2.1) given by

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{f}\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big),\\ \boldsymbol{y}(t) &= \boldsymbol{h}\big(\boldsymbol{x}(t)\big),\end{aligned}$$

with state $\boldsymbol{x}$ taking values $\boldsymbol{x}(t) \in \mathscr{D} \subseteq \mathbb{R}^N$, input $\boldsymbol{u}$ taking values $\boldsymbol{u}(t) \in \mathscr{U} \subseteq \mathbb{R}^p$, and output $\boldsymbol{y}$ taking values $\boldsymbol{y}(t) \in \mathbb{R}^q$. According to Definition 2.1, the system (2.1) is dissipative with respect to a supply rate $s(\boldsymbol{u},\boldsymbol{y})$ if there exists a positive semi-definite storage function $V(\boldsymbol{x}) \geq 0$ that satisfies the condition (2.2), namely that

$$\dot{V}(\boldsymbol{x}(t)) - s(\boldsymbol{u},\boldsymbol{y}) \leq 0$$

along the trajectory of $\boldsymbol{x}(t)$ given by these dynamics. The conclusion of Chapter 2 was that if $\boldsymbol{f}$ and $\boldsymbol{h}$ are polynomial (or polynomial–rational), SOS programming techniques may be used to find appropriate storage functions V that certify the system's dissipativity.

The primary application of the dissipativity-certifying techniques described in Chapter 2 was given in Chapter 3 as the estimation of the change in output trajectory due to a structural perturbation (i.e. model reduction) or parametric perturbation. As discussed in Section 3.4.3, the SOS approach to estimating model reduction error can quickly become impractical as the dimension of the state space increases. In the model reduction setting, it can be safely assumed that the nominal system dimension N is large. Hence, even the most drastic model reduction cannot overcome the fact that the resulting Program 3.3, which seeks to certify the dissipativity of an error system containing $(N+\tilde{N}) > N$ variables, will quickly become impractical to solve. We will show in this chapter that decomposition is a powerful strategy for making a large SOS program of this type more tractable.

Furthermore, the techniques introduced in Chapter 3 for error estimation were concerned with only autonomous models of the form (3.1). However, the dissipativity analysis in Chapter 2 allowed for external inputs to the system, the values of which are constrained to lie in $\mathcal{U} \subseteq \mathbb{R}^p$. When bounding error, the required supply rate is $s(\boldsymbol{u}, \boldsymbol{y}) = -|\boldsymbol{y} - \tilde{\boldsymbol{y}}|^2$. However, it may be that the non-autonomous system is not dissipative with respect to this supply rate. We will show in this chapter that, in this case, dissipativity may be certified by using a different constraint on the trajectory of $\boldsymbol{u}$ that 'imports' more information about the upstream input into the SOS program.

Finally, it was discussed at the end of Chapter 6 that restricting the analysis of signal propagation in layered networks to steady-state responses causes a great deal of information to be lost. This chapter will progress towards rectifying this, by considering how the entire trajectory of a subsystem can respond to a direct perturbation. This response propagates across all subsystems to affect the integrated system output. Our goal is to quantify to what extent the downstream subsystems are affected by the response of the upstream subsystem.

Each of these applications of dissipativity analysis provides the motivation for exploiting a system's structure to construct storage functions using SOS programming. The following section first considers a conventional strategy for decomposing the certification of dissipativity. We then extend this method, for the special case of a cascade system, to introduce Structured Storage Functions (SSFs) and the SOS programs required to construct them. Following the procedure outlined in Chapter 3, SSFs can be applied to certify the dissipativity of a cascaded error system. In Section 7.4 we provide examples which demonstrate the value of this technique to make layered dynamic parameter sensitivity analysis and model reduction error estimation more computationally tractable.

7.2 Structured Storage Functions

There have been a number of approaches to the dissipativity and Lyapunov analysis of structured systems [8, 10, 128, 129, 212]. A common feature of these strategies is that the dissipativity of each subsystem is certified with respect to its own supply rate. This supply rate is usually given, although it may be possible to algorithmically select each subsystem's supply rate [128, 129]. In either case, the decomposed analysis is designed such that a linear combination of the subsystem storage functions suffices as a storage function for the integrated system. This section seeks to extend this strategy, thereby allowing a more flexible range of possible storage functions.

The scope of this chapter will be limited to autonomous systems (2.1) that can be decomposed into cascades of subsystems. First, an autonomous upstream system is defined with state $\boldsymbol{x}_1$ taking values $\boldsymbol{x}_1(t) \in \mathscr{D}_1 \subseteq \mathbb{R}^{N_1}$ such that

$$\dot{\boldsymbol{x}}_1(t) = \boldsymbol{f}_1\big(\boldsymbol{x}_1(t)\big), \quad \boldsymbol{z}_1(t) = \boldsymbol{h}_1\big(\boldsymbol{x}_1(t)\big). \tag{7.1a}$$

A downstream system, with state $\boldsymbol{x}_2$ taking values $\boldsymbol{x}_2 \in \mathscr{D}_2 \subseteq \mathbb{R}^{N_2}$ and with dynamics

$$\dot{\boldsymbol{x}}_2(t) = \boldsymbol{f}_2\big(\boldsymbol{x}_2(t), \boldsymbol{u}_2(t)\big), \quad \boldsymbol{z}_2(t) = \boldsymbol{h}_2\big(\boldsymbol{x}_2(t)\big), \tag{7.1b}$$

is also defined, for input $\boldsymbol{u}_2$. The outputs of the upstream and downstream systems are respectively identified as the downstream input and overall system output, such that

$$\boldsymbol{u}_2(t) = \boldsymbol{z}_1(t), \tag{7.1c}$$

$$\boldsymbol{y}(t) = \boldsymbol{z}_2(t); \tag{7.1d}$$

so that the output of the upstream system (7.1a) drives the downstream system (7.1b).

7.2.1 Linear combinations

The fundamental task of dissipativity analysis is to construct a storage function $V(\boldsymbol{x})$ that certifies the dissipativity of the integrated system (2.1) with respect to a given supply rate $s(\boldsymbol{y})$. The established approach taken in order to decompose this task, in this case, is to construct $V(\boldsymbol{x})$ as the linear combination of subsystem storage functions.

Suppose that two supply rates $s_1(\boldsymbol{z}_1)$ and $s_2(\boldsymbol{u}_2, \boldsymbol{z}_2)$ are given, or can be found, that satisfy the constraint $s_1(\boldsymbol{z}_1) + s_2(\boldsymbol{u}_2, \boldsymbol{z}_2) \leq s(\boldsymbol{y})$. The two equations (7.1c)–(7.1d) defining the relationships between subsystem inputs and outputs are normally used to show this inequality. Each s_i is a supply rate that corresponds to a subsystem in the decomposed dynamics (7.1). For each, we aim to construct a storage function $V_i(\boldsymbol{x}_i) \geq 0$ satisfying $\dot{V}_i - s_i \leq 0$, according to either the upstream dynamics (7.1a) for $i = 1$ or the downstream dynamics (7.1b) for $i = 2$. If these storage functions can be found, then it is simple to show that the linear combination

$$V(\boldsymbol{x}_1, \boldsymbol{x}_2) = V_1(\boldsymbol{x}_1) + V_2(\boldsymbol{x}_2) \tag{7.2}$$

is a storage function certifying the dissipativity of the entire autonomous system (7.1) with respect to $s(\boldsymbol{y})$.

For the special case of cascaded subsystems considered in this chapter, a simple strategy for selecting supply rates s_i is to define the downstream supply rate $s_2(\boldsymbol{u}_2, \boldsymbol{z}_2) = s(\boldsymbol{z}_2) - s_1(\boldsymbol{u}_2)$ for an arbitrary upstream supply rate $s_1(\boldsymbol{z}_1)$. The relations (7.1c)–(7.1d) clearly imply that $s(\boldsymbol{y}) = s_1(\boldsymbol{z}_1) + s_2(\boldsymbol{u}_2, \boldsymbol{z}_2)$, and hence $V(\boldsymbol{x}_1, \boldsymbol{x}_2) = V_1(\boldsymbol{x}_1) + V_2(\boldsymbol{x}_2)$ is a valid storage function if each V_i certifies the dissipativity of subsystem i with respect to s_i.

7.2.2 Introducing Structured Storage Functions

Above, we described one method for exploiting the cascade structure of (7.1) to construct V indirectly as the sum of two subsystem storage functions $V = V_1 + V_2$. We will now introduce the concept of a *Structured Storage Function* (SSF), of which the linear combination (7.2) above is a special case.

Proposition 7.1. *Suppose that the upstream system* (7.1a) *is dissipative with respect to the supply rate* $s_1(\boldsymbol{z}_1)$, *with storage function* $V_1(\boldsymbol{x}_1) \geq 0$ *satisfying*

$$\frac{\mathrm{d}V_1\big(\boldsymbol{x}_1(t)\big)}{\mathrm{d}t} - s_1\big(\boldsymbol{z}_1(t)\big) \leq 0. \tag{7.3a}$$

Suppose also that there exist two positive semi-definite functions $W_0(\boldsymbol{x}_2) \geq 0$ *and* $W_1(\boldsymbol{x}_2) \geq 0$ *such that*

$$\frac{\mathrm{d}W_0\big(\boldsymbol{x}_2(t)\big)}{\mathrm{d}t} + \frac{\mathrm{d}W_1\big(\boldsymbol{x}_2(t)\big)}{\mathrm{d}t} V_1\big(\boldsymbol{x}_1(t)\big) + W_1\big(\boldsymbol{x}_2(t)\big) s_1\big(\boldsymbol{u}_2(t)\big) - s\big(\boldsymbol{z}_2(t)\big) \leq 0. \tag{7.3b}$$

Then the positive semi-definite function

$$V(\boldsymbol{x}_1, \boldsymbol{x}_2) = W_0(\boldsymbol{x}_2) + W_1(\boldsymbol{x}_2) V_1(\boldsymbol{x}_1) \tag{7.4}$$

is a storage function certifying the dissipativity of the cascade system (7.1) *with respect to the global storage function* $s(\boldsymbol{y})$.

Proof. Differentiating V as given by (7.4) results in

$$\begin{aligned} \dot{V} - s &= \dot{W}_0 + \dot{W}_1 V_1 + W_1 \dot{V}_1 - s \\ &\leq \dot{W}_0 + \dot{W}_1 V_1 + W_1 s_1 - s \\ &\leq 0, \end{aligned}$$

for (2.2), where the final inequality is given by (7.3b). □

The function V given by (7.4) is an example of an SSF, which will be defined more comprehensively below. Nevertheless, it can be observed from this example that SSFs will allow for further flexibility beyond that possible through taking linear combinations of upstream and downstream storage functions. The key distinction is that the contribution $V_1\big(\boldsymbol{x}_1(t)\big)$ of the upstream function to V is scaled by a time-varying factor $W_1\big(\boldsymbol{x}_2(t)\big)$. Of course, if we impose that $W_1 = \gamma^2$ is constant for all $\boldsymbol{x}_2$, then we recover the linear combination case above, since W_0 certifies the dissipativity of the downstream system (7.1b) with respect to $s - \gamma^2 s_1$. However, allowing non-constant W_1 enables a greater number of possible SSFs V to be constructed that reflect the underlying cascade structure of the system.

7.2.3 Cascaded information flow

Structured Storage Functions can be further generalised in order to 'import' more information from the upstream system. The following corollary of Proposition 7.1 relies on the parallel dissipativity analysis of the upstream system with respect to K different supply rates.

Corollary 7.2. *Suppose that the upstream system* (7.1a) *is dissipative with respect to each of the K supply rates $s_k(\boldsymbol{x}_1)$, with storage functions $V_k(\boldsymbol{x}_1) \geq 0$ satisfying*

$$\frac{\mathrm{d}V_k\big(\boldsymbol{x}_1(t)\big)}{\mathrm{d}t} - s_k\big(\boldsymbol{z}_1(t)\big) \leq 0, \tag{7.5a}$$

for each $k = 1, \ldots, K$. Suppose also that there exist $K+1$ positive semi-definite functions $W_k(\boldsymbol{x}_2) \geq 0$, for $k = 0, 1, \ldots, K$, such that

$$\frac{\mathrm{d}W_0\big(\boldsymbol{x}_2(t)\big)}{\mathrm{d}t} + \sum_{k=1}^{K} \left[\frac{\mathrm{d}W_k\big(\boldsymbol{x}_2(t)\big)}{\mathrm{d}t} V_k\big(\boldsymbol{x}_1(t)\big) + W_k\big(\boldsymbol{x}_2(t)\big) s_k\big(\boldsymbol{u}_2(t)\big) \right] - s\big(\boldsymbol{z}_2(t)\big) \leq 0. \tag{7.5b}$$

Then the positive semi-definite function

$$V(\boldsymbol{x}_1, \boldsymbol{x}_2) = W_0(\boldsymbol{x}_2) + \sum_{k=1}^{K} W_k(\boldsymbol{x}_2) V_k(\boldsymbol{x}_1) \tag{7.6}$$

is a storage function certifying the dissipativity of the cascade system (7.1) *with respect to the global storage function $s(\boldsymbol{y})$.*

Proof. The proof of this corollary exactly mirrors that of Proposition 7.1, where we apply all K instances of (7.5a) to the derivative of V as defined by (7.6). □

Setting $K = 1$ recovers the example of the SSF given in Proposition 7.1. However, allowing $K > 1$ allows the following general definition of an SSF in the case of a cascade system.

Definition 7.3 A function V of the form (7.6) satisfying (7.5) for $K \geq 1$ is a *Structured Storage Function* of order K. ◊

This definition of an SSF begs the question of exactly which supply rates to use in the upstream analysis (7.5a). It appears that the answer to this question is context-dependent, as will be demonstrated by the example applications given in this chapter. In Section 7.3, the SSF framework is applied to the closely-related problems of parameter sensitivity analysis and model reduction error estimation in the context of large-scale, layered biomolecular networks.

7.2.4 SOS implementation

Before applying this framework to the error/sensitivity estimation problem, this section will discuss the implementation of the SSF approach for cascaded systems (7.1) through SOS programming. For simplicity, it will be assumed that the vector fields $\boldsymbol{f}_i$ and output maps $\boldsymbol{h}_i$ of each subsystem are polynomials. However, similarly to the case discussed in Section 2.3.2, a small adaptation to the following arguments will deal with the case of rational polynomial $\boldsymbol{f}_i$ or $\boldsymbol{h}_i$.

Recall also the technique described in Section 2.3.1 where the sets of state values and input values are estimated by a semi-algebraic set $\mathscr{D} \times \mathscr{U} \subseteq \Phi$ defined by the R polynomial inequalities $\phi_i(\boldsymbol{x}, \boldsymbol{u}) \leq 0$, where $i = 1, \ldots, R$. Similarly, the states $\boldsymbol{x}_i$ of each subsystem of the cascade dynamics (7.1) take values $\boldsymbol{x}_i(t) \in \mathscr{D}_i$ for $i = 1, 2$. Furthermore, the input $\boldsymbol{u}_2 = \boldsymbol{z}_1$ to the downstream system takes values $\boldsymbol{u}_2(t) \in \mathscr{U}_2 = \{\boldsymbol{h}_1(\boldsymbol{x}_1) \mid \boldsymbol{x}_1 \in \mathscr{D}_1\}$. Assume that each of these spaces can be estimated such that

$$\mathscr{D}_1 \subseteq \{\boldsymbol{x}_1 \in \mathbb{R}^{N_1} \mid \phi_{1,r}(\boldsymbol{x}_1) \le 0 \text{ for } r = 1, \dots, R_1\} = \Phi_1,$$
$$\mathscr{D}_2 \times \mathscr{U}_2 \subseteq \{(\boldsymbol{x}_2, \boldsymbol{u}_2) \in \mathbb{R}^{N_2+p_2} \mid \phi_{2,r}(\boldsymbol{x}_2, \boldsymbol{u}_2) \le 0 \text{ for } r = 1, \dots, R_2\} = \Phi_2.$$

The second of these bounds allows for constraints on the values of $\boldsymbol{u}_2(t) = \boldsymbol{z}_1(t) \in \mathscr{U}_2$, if such a condition can be found, thereby importing a limited degree of information from the upstream system.

To construct an SSF of the form (7.6) we must find K upstream polynomial storage functions $V_k(\boldsymbol{x}_1) \ge 0$, for $k = 1, \dots, K$, each satisfying (7.5a). We must also find $(K+1)$ polynomials $W_k(\boldsymbol{x}_2) \ge 0$, for $k = 0, 1, \dots, K$, which together satisfy (7.5b). This search can be implemented by a total of $(K+1)$ SOS programs, which can all be run in parallel, such that the resulting outputs of each can be combined into a feasible SSF.

Program 7.4 (Upstream Program $k \in \{1, 2, \dots, K\}$) Consider the state $\boldsymbol{x}_1$ and output $\boldsymbol{z}_1$ with dynamics given by (7.1a) for polynomial $\boldsymbol{f}_1$ and $\boldsymbol{h}_1$, and the estimate $\mathscr{D}_1 \subseteq \Phi_1$ on the state space. For the fixed value of k, define an SOS polynomial decision variable $V_k(\boldsymbol{x}_1) \in \Sigma[\boldsymbol{x}_1]$, and a further R_1 SOS polynomial decision variables $\sigma_{k,r}(\boldsymbol{x}_1) \in \Sigma[\boldsymbol{x}_1]$ for $r = 1, \dots, R_1$. For the given supply rate $s_k(\boldsymbol{z}_1)$, find feasible V_k and $\sigma_{k,r}$ subject to the constraint

$$-\left(\frac{\mathrm{d}V_k}{\mathrm{d}t}(\boldsymbol{x}_1) - s_k(\boldsymbol{z}_1)\right) + \sum_{r=1}^{R_1} \sigma_{k,r}(\boldsymbol{x}_1)\phi_{1,r}(\boldsymbol{x}_1) \in \Sigma[\boldsymbol{x}_1] \tag{7.7}$$

to return a storage function $V_k(\boldsymbol{x}_1)$ certifying the dissipativity of the upstream system with respect to $s_k(\boldsymbol{z}_1)$. ◊

Program 7.5 (Downstream Program) Consider the state $\boldsymbol{x}_2$ and output $\boldsymbol{z}_2 = \boldsymbol{y}$ with dynamics given by (7.1b) for polynomial $\boldsymbol{f}_2$ and $\boldsymbol{h}_2$, and the estimate $\mathscr{D}_2 \times \mathscr{U}_2 \subseteq \Phi_2$ on the state space and the set of input values. For the additional K-dimensional auxiliary variable $\boldsymbol{w} = (w_1, \dots, w_K)^T$ define the following $R_2 + 2K + 1$ SOS polynomial decision variables:

$$W_k(\boldsymbol{x}_2) \in \Sigma[\boldsymbol{x}_2] \text{ for } k = 0, 1, \dots, K,$$
$$\tau_k(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}) \in \Sigma[\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}] \text{ for } k = 1, \dots, K,$$
$$\rho_r(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}) \in \Sigma[\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}] \text{ for } r = 1, \dots, R_2.$$

Given a supply rate $s(\boldsymbol{y}) = s(\boldsymbol{z}_2)$, find feasible W_k, τ_k, and ρ_r subject to the constraint

$$\begin{aligned} &-\left(\frac{\mathrm{d}W_0(\boldsymbol{x}_2)}{\mathrm{d}t} + \sum_{k=1}^{K}\left(\frac{\mathrm{d}W_k(\boldsymbol{x}_2)}{\mathrm{d}t} w_k + s_k(\boldsymbol{u}_2)W_k(\boldsymbol{x}_2)\right) - s(\boldsymbol{z}_2)\right) \\ &+ \sum_{r=1}^{N_2} \rho_r(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w})\phi_{2,r}(\boldsymbol{x}_2, \boldsymbol{u}_2) - \sum_{k=1}^{K} \tau_k(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}) w_k \in \Sigma[\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w}] \end{aligned} \tag{7.8}$$

to return downstream functions $W_k(\boldsymbol{x}_2)$ for $k = 0, 1, \dots, K$. ◊

Proposition 7.6. *Using the outputs V_k, for $k = 1, \dots, K$, of each instance of Program 7.4, and the outputs W_k, for $k = 0, 1, \dots, K$ of Program 7.5, construct a function V according to the formula* *in* (7.6). *The polynomial V is an SSF that certifies the dissipativity of the interconnected cascade* (7.1) *with respect to the global supply rate $s(\boldsymbol{y})$.*

Proof. Each V_k is constructed to satisfy the constraint (7.7). For each $r = 1, \dots, R_1$, the polynomial $\phi_{1,r}(\boldsymbol{x}_1) \le 0$ for all $\boldsymbol{x}_1 \in \mathscr{D}_1$. It follows from (7.7) that the polynomials V_k satisfy (7.5a) in $\mathscr{D}_1$, and are therefore storage functions certifying the dissipativity of the upstream system (7.1a) with respect to each supply rate $s_k(\boldsymbol{z}_1)$.

In place of the K additional variables $\boldsymbol{w} = (w_k)$, we can substitute the upstream storage functions $w_k(t) = V_k\big(\boldsymbol{x}_1(t)\big)$ into (7.8). Hence the downstream condition (7.5b) holds for $(\boldsymbol{x}, \boldsymbol{u}) \in \mathscr{D}_2 \times \mathscr{U}_2$ and $w_k = V_k \in \Sigma[\boldsymbol{x}_1]$. Since both expressions in (7.5) hold, the polynomial $V = W_0 + \sum_{k=1}^{K} W_k V_k$ given by (7.6) is an SSF. □

Note that, rather than directly including $V_k(\boldsymbol{x}_1)$ as known functions in the downstream Program 7.5, we instead have extended the state space $\boldsymbol{x}_2$ with the K auxiliary variables $\boldsymbol{w} = (w_1, \dots, w_K)^T$, constrained such that $w_k \ge 0$ for all k. This decouples all $(K+1)$ SOS programs so that each can be run independently and in parallel, resulting in a significant computational saving. The resulting $2K+1$ SOS polynomials V_k, W_k (for $k = 1, \dots, K$), and W_0 are then combined, once all have been found separately, to give V. The following discussion explores the computational savings of this approach in more detail.

7.2.5 Computational saving

An important goal for the decomposition of dissipativity analysis is to speed computation. Section 2.4 discussed how an SOS programming approach to certifying dissipativity quickly becomes intractable as either the degree of the polynomial in (2.5b) or the dimension N of the state space become large. This section considers the conditions under which the SSF approach can provide significant computational savings.

We assume that the system (7.1) is not uncertain, has polynomial dynamics, and is autonomous. Without exploiting structure, the required SOS program for certifying dissipativity is given by Program 2.11. There are $R+1$ SOS decision variables ($V(\boldsymbol{x})$ and $\sigma_i(\boldsymbol{x})$, where $i = 1, \dots, R$) defined by Program 2.11, where R is the number of polynomials $\phi_i(\boldsymbol{x})$ defining the estimate $\Phi \supseteq \mathscr{D}$. Each of these is a polynomial of even degree $2d_i$ in N variables, for $i = 0, 1, \dots, R$. The number of decision variables in the resulting semi-definite program contributed by each polynomial is indicated in Table 2.1, and grows extremely rapidly with both N and d_i.

In contrast, we now consider the structured approach combining K instances of Program 7.4 with Program 7.5. Each instance of Program 7.4 for fixed k is an SOS program seeking $R_1 + 1$ SOS decision variables: $V_k(\boldsymbol{x}_1)$, and $\sigma_{k,r}$ for $r = 1, \dots, R_1$. Each of these polynomials takes arguments $\boldsymbol{x}_1$ in $N_1 < N$ dimensions. Also, since $\mathscr{D}_1$ is in $N_1 < N$ dimensions, it is likely that the estimate $\Phi_1 \supseteq \mathscr{D}_1$ can be defined by $R_1 < R$ polynomials. This implies that the computation of each of the upstream programs will be significantly cheaper than the full program.

In addition to the cheaper computation of Program 7.4, we also consider the computational cost of Program 7.5. There are $R_2 + 2K + 1$ SOS decision variables in this program: $W_0(\boldsymbol{x}_2)$, $W_k(\boldsymbol{x}_2)$ and $\tau_k(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w})$ for $k = 1, \dots, K$, and $\rho_r(\boldsymbol{x}_2, \boldsymbol{u}_2, \boldsymbol{w})$ for $r = 1, \dots, R_2$. Since $\boldsymbol{u}_2$ and $\boldsymbol{w}$ also appear in the arguments of some decision variables, the resulting polynomials are SOS polynomials in N_2+

$N_{\boldsymbol{u}_2}+K$ variables, where $N_{\boldsymbol{u}_2}$ denotes the number of inputs to the downstream system. Therefore, to ensure that the decomposed strategy of this chapter provides computational savings over the standard approach of Program 2.11, one possible heuristic is that the inequality $N_{\boldsymbol{u}_2}+K<N_1$ should hold as strongly as possible.

Each term on the left-hand side of this inequality can be minimised. We should ensure that the number of inputs from upstream to downstream (that is, $N_{\boldsymbol{u}_2}$) is as small as possible.[1] Another way to reduce computational burden is to make the number K of upstream analyses small. This requirement implies that the K upstream supply rates s_k should be chosen to be as informative about the upstream system as possible; this is discussed in more detail in Section 7.6.1 below.

For example, consider a cascade of two systems, chosen such that $N_1=6$ and $N_2=4$, so that $N=10$. Suppose the upstream output $\boldsymbol{z}_1=\boldsymbol{u}_2$ that drives the downstream system is a scalar, so that $N_{\boldsymbol{u}_2}=1$. Assuming $K=1$, Program 2.11 in 10-dimensional space can be decomposed into $K+1=2$ SOS programs. The upstream computation in Program 7.4 seeks polynomials of $N_1=6$ variables, while the downstream computation in Program 7.5 seeks polynomials of $N_2+N_{\boldsymbol{u}_2}+K=4+1+1=6$ variables also. Given that the computational burden of SOS programs increases extremely rapidly with state dimension, this means that the decomposed strategy will provide a significant saving when certifying the dissipativity of (7.1). Section 7.4 will demonstrate the benefit of this speed-up on two example BRN models.

7.2.6 Optimisation

Suppose that an objective function on V is also required. When decomposing the SOS program, the objective function must also be decomposed into each of the upstream programs and the downstream program, at the potential cost of some suboptimality. Without the objective function, all of the K instances of Program 7.4, together with Program 7.5, are decoupled. However, when an objective function is overlaid with the feasibility constraints given in each of these SOS programs, the optimisation problem needs to be carefully decomposed to balance the trade-off between the potential conservatism of the optimised result and computational speed [47].

It will be shown below that decomposing the estimation of reduction error and parameter sensitivity admits a heuristic, greedy approach that balances these competing requirements. First, each of the K instances of Program 7.4 can be computed in parallel with its own objective function, returning optimal V_k for $k=1,\dots,K$. The optimal values of the K upstream objective functions will be used to define an objective for Program 7.5. This implies that the downstream analysis is no longer decoupled from the upstream analysis, but ensures that the resulting storage functions provide a tight upper estimate of the optimal upper bound. Examples of this approach can be seen in Programs 7.10 and 7.11 below.

7.3 Cascaded Perturbations

This section applies the decomposed dissipativity analysis above to the error system problem given in Chapter 3. Similarly to the definition of the nominal and perturbed systems in (3.1), we

[1] This criterion is similar to that proposed in Section 6.5.1 of minimising the rank of the inter-layer communication matrix $\boldsymbol{A}_{lk}$, giving the minimal number of inputs from layer k to layer l.

can define a nominal and perturbed cascade system with dynamics

$$\begin{aligned} \text{Nominal upstream: } &\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1;\boldsymbol{\pi}_1), \quad \boldsymbol{z}_1 = \boldsymbol{h}_1(\boldsymbol{x}_1), \quad \boldsymbol{x}_1(0) = \boldsymbol{x}_{1,0}; \\ \text{Perturbed upstream: } &\dot{\tilde{\boldsymbol{x}}}_1 = \tilde{\boldsymbol{f}}_1(\tilde{\boldsymbol{x}}_1;\tilde{\boldsymbol{\pi}}_1), \quad \tilde{\boldsymbol{z}}_1 = \tilde{\boldsymbol{h}}_1(\tilde{\boldsymbol{x}}_1), \quad \tilde{\boldsymbol{x}}_1(0) = \boldsymbol{\chi}_1(\boldsymbol{x}_{1,0}), \end{aligned} \tag{7.9a}$$

where each output $\boldsymbol{z}_1 = \boldsymbol{u}_2$ and $\tilde{\boldsymbol{z}}_1 = \tilde{\boldsymbol{u}}_2$ forms an input to the downstream systems

$$\begin{aligned} \text{Nominal downstream: } &\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2,\boldsymbol{u}_2;\boldsymbol{\pi}_2), \quad \boldsymbol{z}_2 = \boldsymbol{h}_2(\boldsymbol{x}_2), \quad \boldsymbol{x}_2(0) = \boldsymbol{x}_{2,0}; \\ \text{Perturbed downstream: } &\dot{\tilde{\boldsymbol{x}}}_2 = \tilde{\boldsymbol{f}}_2(\tilde{\boldsymbol{x}}_2,\tilde{\boldsymbol{u}}_2;\tilde{\boldsymbol{\pi}}_2), \quad \tilde{\boldsymbol{z}}_2 = \tilde{\boldsymbol{h}}_2(\tilde{\boldsymbol{x}}_2), \quad \tilde{\boldsymbol{x}}_2(0) = \boldsymbol{\chi}_2(\boldsymbol{x}_{2,0}). \end{aligned} \tag{7.9b}$$

The two outputs of the cascade are denoted $\boldsymbol{y} = \boldsymbol{z}_2$ and $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{z}}_2$. As with (3.1), this notation captures many of the possible perturbations to the cascade system. For example, parameter sensitivity analysis is the case where $\boldsymbol{\pi}_i \neq \tilde{\boldsymbol{\pi}}_i$ for $i = 1$ or $i = 2$. The task of this section, as in Chapter 3, is to find an upper bound on the $\mathscr{L}_2$ norm of the difference in outputs $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|$ incurred by structural or parametric perturbations. However, unlike Chapter 3, we will not consider the worst-case response, and instead assume that the initial conditions $\boldsymbol{x}_i(0)$ and $\tilde{\boldsymbol{x}}_i(0)$ of each subsystem $i = 1, 2$ are fixed. Nevertheless, the results below can be easily extended in the manner described in Section 3.1.3 to consider worst-case responses over sets $\mathscr{X}_{i,0}$ of initial conditions for each subsystem.

7.3.1 Incremental gain

We will first consider the case where the downstream system is not directly perturbed, so that the vector fields $\boldsymbol{f}_2 = \tilde{\boldsymbol{f}}_2$, the output maps $\boldsymbol{h}_2 = \tilde{\boldsymbol{h}}_2$, the parameters $\boldsymbol{\pi}_2 = \tilde{\boldsymbol{\pi}}_2$, and the initial conditions $\boldsymbol{x}_2(0) = \tilde{\boldsymbol{x}}_2(0)$ are all equal. The only disturbance to the downstream system arises from the response of the upstream system, where $\boldsymbol{u}_2 \neq \tilde{\boldsymbol{u}}_2$. In this case, the concept of incremental gain [24, 163] can be applied to estimate an upper bound on the resulting output difference $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2$.

Suppose there exists $\gamma \geq 0$ such that the combined downstream system (7.9b) is dissipative with respect to the supply rate

$$s_\gamma(\boldsymbol{u}_2, \tilde{\boldsymbol{u}}_2, \boldsymbol{z}_2, \tilde{\boldsymbol{z}}_2) = \gamma^2 |\boldsymbol{u}_2 - \tilde{\boldsymbol{u}}_2|^2 - |\boldsymbol{z}_2 - \tilde{\boldsymbol{z}}_2|^2,$$

and define the set Γ as the set of all such γ. Then the incremental gain $\gamma^\star$ of the system is defined as $\gamma^\star = \inf(\Gamma)$. For a fixed feasible $\gamma \in \Gamma$, the corresponding storage function will be denoted $W_0(\boldsymbol{x}_2, \tilde{\boldsymbol{x}}_2)$. Substituting s_γ into the dissipativity inequality (2.2) implies that

$$|\boldsymbol{z}_2 - \tilde{\boldsymbol{z}}_2|^2 \leq \gamma^2 |\boldsymbol{u}_2 - \tilde{\boldsymbol{u}}_2|^2 - \frac{\mathrm{d}W_0\big(\boldsymbol{x}_2(t), \tilde{\boldsymbol{x}}_2(t)\big)}{\mathrm{d}t}. \tag{7.10}$$

Then, similarly to the error estimation technique in Section 3.1, we can integrate this inequality across the entire time horizon $t \geq 0$ to deduce the linear inequality

$$\|\boldsymbol{z}_2 - \tilde{\boldsymbol{z}}_2\|^2 \leq \gamma^2 \|\boldsymbol{u}_2 - \tilde{\boldsymbol{u}}_2\|^2 + W_0\big(\boldsymbol{x}_2(0), \tilde{\boldsymbol{x}}_2(0)\big), \tag{7.11}$$

where it is assumed, as in Chapter 2, that $\boldsymbol{x}_2, \tilde{\boldsymbol{x}}_2 \to \boldsymbol{0}$ and $W_0(\boldsymbol{0}, \boldsymbol{0}) = 0$. This provides an upper bound on the $\mathscr{L}_2$ norm of the change in output as a linear function of the $\mathscr{L}_2$ norm of the change in input.

Hence, the technique of incremental gain enables a decomposed approach to perturbation analysis through dissipativity. Suppose that a storage function $V_1(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1)$ exists such that the upstream system (7.9a) is dissipative with respect to $s_1(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_1) = |\boldsymbol{z}_1 - \tilde{\boldsymbol{z}}_1|^2$. The function $V(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1, \boldsymbol{x}_2, \tilde{\boldsymbol{x}}_2) = \gamma^2 V_1 + W_0$ is therefore a storage function proving the dissipativity of the entire system with respect to $s = -|\boldsymbol{z}_2 - \tilde{\boldsymbol{z}}_2|^2$, and the error bound

$$\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 \leq \gamma^2 V_1\big(\boldsymbol{x}_1(0), \tilde{\boldsymbol{x}}_1(0)\big) + W_0\big(\boldsymbol{x}_2(0), \tilde{\boldsymbol{x}}_2(0)\big) = V|_{t=0}$$

is implied by (7.11). The factor γ^2 quantifies how the upstream perturbation, estimated by $V\big(\boldsymbol{x}_1(0), \tilde{\boldsymbol{x}}_1(0)\big)$, propagates downstream. The function V is clearly an SSF of the form (7.6), with $K = 1$ and constant $W_1 = \gamma^2$.

There may be some cases for which no γ exists such that the dissipativity of the downstream system (7.9b) with respect to s_γ can be shown. Section 7.4 will illustrate two such cases, and demonstrate how to overcome these problems, using the more generic SSF formulation with $K \geq 1$ and non-constant $W_k(\boldsymbol{x}_2)$ for $k = 1, \ldots, K$.

7.3.2 SSFs and error signal propagation

In Section 7.2 we showed how SSFs provide a method for decomposing the dissipativity analysis of large-scale nonlinear systems, containing the summed storage function approach as a special case. Indeed, the incremental gain approach is essentially equivalent to the summed storage function approach for a particular choice of upstream storage function, and assuming that the downstream subsystem is not perturbed.

For any perturbation to either the upstream or downstream subsystem, the SOS programs that can be used to bound the response $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|$ are Program 7.4 and Program 7.5, applied to the upstream system (7.9a) and downstream system (7.9b) respectively.

Proposition 7.7. *Consider each output V_k, $k = 1, \ldots, K$ of the K instances of Program 7.4 applied to* (7.9a)*, and for the $K+1$ outputs W_k, $k = 0, \ldots, K$ of Program 7.5 applied to* (7.9b). *Then, for fixed initial conditions $\boldsymbol{x}_i(0)$ and $\tilde{\boldsymbol{x}}_i(0)$ for each subsystem $i = 1, 2$, the inequality*

$$\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 \leq W_0\big(\boldsymbol{x}_2(0), \tilde{\boldsymbol{x}}_2(0)\big) + \sum_{k=1}^{K} W_k\big(\boldsymbol{x}_2(0), \tilde{\boldsymbol{x}}_2(0)\big) V_k\big(\boldsymbol{x}_1(0), \tilde{\boldsymbol{x}}_1(0)\big)$$

is an upper bound, derived from the structured storage function $V = W_0 + \sum_{k=1}^{K} W_k V_k$, for the response of the nominal system to structural or parametric perturbations.

We will show later in this chapter that the flexibility of SSFs over the incremental gain approach has a number of benefits. First, it is much more widely applicable, since we can deal with perturbations to the downstream system, rather than simply to its input. Furthermore, when incremental gain methods cannot bound the output response, using non-constant SOS variables W_k and different (or additional) upstream supply rates $s_k(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_1)$ may be enough to enable dissipativity to be shown. Finally, even when a bound can be found through using incremental gain, it may be the case that the SSF-derived bound is tighter, since it includes constant $W_1 = \gamma^2$ as a special case.

7.4 Example: Layered Sensitivity

This section will demonstrate how to decompose the dissipativity analysis of the cascaded BRN system

$$X + E \underset{\beta_{-1}}{\overset{\beta_1}{\rightleftharpoons}} C \xrightarrow{\beta_2} Y + E, \tag{7.12a}$$

$$Y + K \xrightarrow{\beta_3} Z, \tag{7.12b}$$

which will also illustrate situations where the more general SSF formulation enables a decomposed strategy when incremental gain approaches fail.

The dynamics of (7.12) can be written in the form (7.1), where it is assumed that the concentration of Z is the output of interest. For the upstream state $\boldsymbol{x}_1 = (x_{1,1}, x_{1,2})$ and (scalar) downstream state x_2, this system can be realised by the ODEs

$$\begin{bmatrix} \dot{x}_{1,1} \\ \dot{x}_{1,2} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1(-x_{1,1} - x_{1,2})(P_0 - x_{1,1}) - \beta_{-1}x_{1,1} \\ \beta_2 x_{1,1} \end{bmatrix}, \qquad z_1 = x_{1,2}, \tag{7.13a}$$

$$\dot{x}_2 = \beta_3 (k_0 - R - x_2)\big(x_0 + y_0 - R + u_2 - x_2\big), \qquad z_2 = x_2, \tag{7.13b}$$

for $u_2 = z_1$ and $y = z_2$, and with initial conditions $\boldsymbol{x}_1(0) = (0, -x_0)$ and $x_2(0) = -R$ for the parameter $R = \min(k_0, x_0 + y_0)$. It can be shown that each state approaches zero as $t \to \infty$. The relation between the dynamical system (7.13) and the reactions (7.12) is shown in Appendix C.

This section will consider perturbations to the parameter β_2 from its nominal value $\beta_2 = 1$ to a new value $\tilde{\beta}_2 > 1$. The steady state of this system will not change, but the transient dynamics will be altered. The following results compare the SOS dissipativity techniqe of Chapter 3 with the decomposed analysis of this chapter.

7.4.1 Parametric perturbations

First consider the unstructured SOS approach, given in Chapter 3, to the parametric perturbation of (7.13). We first compare the output trajectory $y = x_2$ of this system to the output $\tilde{y} = \tilde{x}_2$ of the perturbed system

$$\begin{bmatrix} \dot{\tilde{x}}_{1,1} \\ \dot{\tilde{x}}_{1,2} \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1(-\tilde{x}_{1,1} - \tilde{x}_{1,2})(P_0 - \tilde{x}_{1,1}) - \beta_{-1}\tilde{x}_{1,1} \\ \tilde{\beta}_2 \tilde{x}_{1,1} \\ \beta_3(k_0 - R - \tilde{x}_2)(x_0 + y_0 - R + \tilde{x}_{1,2} - \tilde{x}_2) \end{bmatrix}, \tag{7.14}$$

where the nominal vector value β_2 has been perturbed to $\tilde{\beta}_2$. The dynamics (7.13) and (7.14) correspond to the form (3.1), with output $y - \tilde{y} = x_2 - \tilde{x}_2$. The following SOS program can be used to find an upper bound on the $\mathscr{L}_2$ norm of this response.

$\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|^2$ bounds	$k_0 = 1$	$k_0 = 3$	$k_0 = 5$
$\tilde{\beta}_2 = 2$	0.0004	0.0930	0.6792
$\tilde{\beta}_2 = 3$	0.0012	0.1865	1.2944

Table 7.1: Upper bounds on parameter sensitivity $\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|^2$ for varying parameter perturbations and varying initial concentrations k_0 of K.

Program 7.8 Define the SOS polynomial decision variables $V(\boldsymbol{x},\tilde{\boldsymbol{x}})$, $\sigma_i(\boldsymbol{x},\tilde{\boldsymbol{x}})$, and $\tilde{\sigma}_j(\boldsymbol{x},\tilde{\boldsymbol{x}}) \in \Sigma[\boldsymbol{x},\tilde{\boldsymbol{x}}]$, for $i, j = 1,\dots,4$. Minimise the upper bound $V(0,-x_0,-R,0,-x_0,-R)$ subject to the SOS constraint

$$-\left[\dot{V} + (x_2 - \tilde{x}_2)^2\right] + \sum_{i=1}^{4} \sigma_i \phi_i + \sum_{j=1}^{4} \tilde{\sigma}_j \tilde{\phi}_j \in \Sigma[\boldsymbol{x},\tilde{\boldsymbol{x}}],$$

where the eight polynomials $\phi_i(\boldsymbol{x},\tilde{\boldsymbol{x}})$ and $\tilde{\phi}_i(\boldsymbol{x},\tilde{\boldsymbol{x}})$ defining an upper estimate on the state space are given by

$$\begin{aligned}
\phi_1 &= x_{1,1}(x_{1,1} - P_0), & \tilde{\phi}_1 &= \tilde{x}_{1,1}(\tilde{x}_{1,1} - P_0),\\
\phi_2 &= x_{1,2}(x_{1,2} + x_0), & \tilde{\phi}_2 &= \tilde{x}_{1,2}(\tilde{x}_{1,2} + x_0),\\
\phi_3 &= x_2(x_2 + R), & \tilde{\phi}_3 &= \tilde{x}_2(\tilde{x}_2 + R),\\
\phi_4 &= -(x_0 + y_0 - R + x_{1,2} - x_2), & \tilde{\phi}_4 &= -(x_0 + y_0 - R + \tilde{x}_{1,2} - \tilde{x}_2).
\end{aligned}$$

◊

The polynomials ϕ_i and $\tilde{\phi}_j$ above have been defined such that $\phi_i \le 0$ and $\tilde{\phi}_j \le 0$ implies that the concentration of each species is non-negative.

We ran each of the SOS programs for the nominal parameter set $(\beta_1, \beta_{-1}, \beta_2, \beta_3, P_0)$ given by $(1,1,1,1,1)$, two perturbed values of $\tilde{\beta}_2 = 2$ and $\tilde{\beta}_2 = 3$, and across three initial conditions $(x_0, y_0, k_0) = (2,2,k_0)$ for three values of $k_0 = 1,3,5$. This required a total of six SOS programs, resulting in the upper bounds on the parameter sensitivity $\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|$ that are displayed in Table 7.1. As in Section 3.2.2, these bounds are extremely close to the simulated output responses (data not shown).

7.4.2 Upstream analysis

In contrast to the example of unstructured dissipativity analysis, the computation can be decomposed by applying the SSF techniques introduced in this chapter. In this example, the perturbation of β_2 to $\tilde{\beta}_2$ applies directly only to the upstream layer's dynamics (7.13a); the downstream dynamics (7.13b) are not directly affected, and instead respond to the perturbation through a change in the nominal input trajectory $u_2 = z_1 = x_{2,1}$ to the perturbed input trajectory $\tilde{u}_2 = \tilde{z}_1 = \tilde{x}_{2,1}$.

In this section, we substitute the nominal upstream dynamics (7.13a) into the upstream error system (7.9a). The dynamics

$$\begin{bmatrix} \dot{\tilde{x}}_{1,1} \\ \dot{\tilde{x}}_{1,2} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1(-\tilde{x}_{1,1} - \tilde{x}_{1,2})(P_0 - \tilde{x}_{1,1}) - \beta_{-1}\tilde{x}_{1,1} \\ \tilde{\beta}_2 \tilde{x}_{1,1} \end{bmatrix},$$

Upstream analysis	$\|z_1 - \tilde{z}_1\|^2$	$\|z_1 + \tilde{z}_1\|^2$
Parameter $\tilde{\beta}_2 = 2$	0.7458	27.31
Parameter $\tilde{\beta}_2 = 3$	1.4328	24.48

Table 7.2: Upper bounds on the output functionals $\|z_1 - \tilde{z}_1\|^2$ and $\|z_1 + \tilde{z}_1\|^2$ measuring the response of the upstream subsystem (7.13a) to two different parameter perturbations.

with perturbed parameter $\tilde{\beta}_2$ and output $\tilde{z}_1 = \tilde{x}_{1,2}$, is used as the perturbed system in (7.9a). Similarly to the combined analysis above, we can attempt to bound any given supply rate $s_k(z_1, \tilde{z}_1)$, where the integer k indexes the set of possible supply rates, through the following instance of Program 7.4.

Program 7.9 For the SOS decision variables $V_k(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1) \in \Sigma[\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1]$, $\sigma_i(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1) \in \Sigma[\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1]$, and $\tilde{\sigma}_j(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1) \in \Sigma[\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1]$, for $i, j = 1, \dots, 2$, we seek to minimise $V_k(0, -x_0, 0, -x_0)$ subject to the SOS constraint

$$-\left[\dot{V}_k - s_k(z_1, \tilde{z}_1)\right] + \sum_{i=1}^{2} \sigma_i \phi_i + \sum_{j=1}^{2} \tilde{\sigma}_j \tilde{\phi}_j \in \Sigma[\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1]$$

for the four polynomials $\phi_i(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1)$ and $\tilde{\phi}_j(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1)$ given above with $i, j = 1, 2$. ◊

For example, if $s_1 = -(z_1 - \tilde{z}_1)^2$, then Program 7.9 bounds the sensitivity of the upstream output z_1 to the parameter perturbation $\tilde{\beta}_2$. Alternatively, the supply rate $s_2 = -(z_1 + \tilde{z}_1)^2$ can be used in Program 7.9 to bound the total size of both systems' outputs. The results for each of these storage functions are given in Table 7.2. Note that, because the nominal and perturbed upstream system's dynamics are both independent of k_0, the varying values of k_0 given in Table 7.1 have no effect on the upstream error.

7.4.3 Incremental gain

The parameter perturbation only affects the upstream layer's dynamics. This means that the response of the downstream layer's output depends only on the change in the input from upstream, from $u_2 = z_1$ to $\tilde{u}_2 = \tilde{z}_1$. This section applies incremental gain techniques, discussed in Section 7.3.1, to estimate the propagation of the upstream response into the downstream system. It will be shown that, for this example, a parameter region exists for which incremental gain techniques fail.

We substitute the downstream layer's dynamics (7.13b), with nominal input u_2 and perturbed input $\tilde{u}_2$ respectively, into the error system (7.9b). Thus this section is concerned with the resulting error system

$$\dot{x}_2 = \beta_3(k_0 - R - x_2)(x_0 + y_0 - R + u_2 - x_2) \tag{7.15a}$$

$$\dot{\tilde{x}}_2 = \beta_3(k_0 - R - \tilde{x}_2)(x_0 + y_0 - R + \tilde{u}_2 - \tilde{x}_2), \tag{7.15b}$$

with initial conditions $x_2(0) = \tilde{x}_2(0) = -R$. The response of the output, measured by $\|x_2 - \tilde{x}_2\|^2$, can be related to the difference $\|u_2 - \tilde{u}_2\|^2$ in the input through the following SOS program for finding the the incremental gain of the downstream layer (7.13b).

Program 7.10 For the SOS decision variables $W_0(x_2, \tilde{x}_2) \in \Sigma[x_2, \tilde{x}_2]$, $\sigma_i(x_2, \tilde{x}_2, u_2, \tilde{u}_2) \in \Sigma[x_2, \tilde{x}_2, u_2, \tilde{u}_2]$, and $\tilde{\sigma}_j(x_2, \tilde{x}_2, u_2, \tilde{u}_2) \in \Sigma[x_2, \tilde{x}_2, u_2, \tilde{u}_2]$, where $i, j = 1, \ldots, 3$, minimise one of the following three objective functions:

OBJ1 The additional constant decision variable γ^2;

OBJ2 The upper bound $\gamma^2 E^2 + W_0(-R, -R)$ over the additional constant decision variable γ^2, where E^2 is a pre-determined upper bound on the upstream error $\| u_2 - \tilde{u}_2 \|^2 \leq E^2$;

OBJ3 $W_0(-R, -R)$, given a pre-defined specific value of γ^2;

subject to the SOS constraint

$$- \left[\dot{W}_0 + (x_2 - \tilde{x}_2)^2 - \gamma^2 (u_2 - \tilde{u}_2)^2 \right] + \sum_{i=1}^{3} \sigma_i \theta_i + \sum_{j=1}^{3} \tilde{\sigma}_j \tilde{\theta}_j \in \Sigma[x_2, \tilde{x}_2, u_2, \tilde{u}_2]$$

where the polynomials $\theta_i(x_2, u_2) = \phi_{i+1}|_{x_{1,2}=u_2}$ and $\tilde{\theta}_j(\tilde{x}_2, \tilde{u}_2) = \tilde{\phi}_{j+1}|_{\tilde{x}_{1,2}=\tilde{u}_2}$ that define the estimate of the state space and input space

$$\mathscr{D}_2 \times \mathscr{U}_2 \subseteq \{(x_2, \tilde{x}_2, u_2, \tilde{u}_2) \mid \theta_i \leq 0,\ \tilde{\theta}_j \leq 0\ \forall i, j = 1, 2, 3\} \tag{7.16}$$

are given by ϕ_{i+1} and $\tilde{\phi}_{j+1}$ defined in Section 7.4.1. ◇

Recall that the results of the upstream dissipativity analysis, given in Table 7.2, are independent of k_0. However, this parameter has an important effect on the downstream dissipativity analysis. This 'incremental gain' SOS program will now be explored further for varying values of $k_0 = 1, 3, 5$, which are representative of three distinct parameter regions.

An excess of K We first consider the case of the dynamics (7.13b) where $x_0 + y_0 < k_0$, so that $R = \min(x_0 + y_0, k_0) = x_0 + y_0$. In terms of the original dynamics (7.12), this corresponds to the situation where the initial concentration of K is in sufficient excess to consume all of the initial mass of Y, plus that which is produced by X. We keep the same parameter values as previously, setting $k_0 = 5$.

Running Program 7.10 with these parameters and objective function OBJ1, the incremental gain is found to be bounded above by $\gamma^2 = 1 + \epsilon$, for a negligibly small value of ϵ. If instead we fix $\gamma^2 = 1 + \epsilon$ for a given small parameter $\epsilon \ll 1$, then optimising objective function OBJ3 results in a minimal value of $W_0(-R, -R) = 0$, correct to seven decimal places. Therefore we can deduce an upper bound of

$$\| x_2 - \tilde{x}_2 \|^2 \leq (1 + \epsilon) \| u_2 - \tilde{u}_2 \|^2$$

in the case when $k_0 = 5$, for arbitrarily small values of ϵ that we can approximate as zero. Thus the bounds on the values of $\| u_2 - \tilde{u}_2 \|^2$ given in the first column of Table 7.2 are also upper bounds on the downstream error. Comparing those values to the third column of Table 7.1, corresponding to $k_0 = 5$, it is clear that there is a degree of conservatism in this bound. This conservatism is the price of decomposing the unstructured storage function V found by Program 7.8 into the SSF form $W_0(x_2, \tilde{x}_2) + \gamma^2 V_1(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1)$ found by the combination of Program 7.9 and Program 7.10.

A scarcity of K We now consider the case of the dynamics (7.13b) where $k_0 < y_0 < x_0 + y_0$, so that $R = \min(x_0 + y_0, k_0) = k_0$. In terms of the original dynamics (7.12), this corresponds to the situation where the initial concentration of K is not sufficient to consume all of the initial concentration of Y, nor any of the additional Y produced from X through the constant enzyme concentration P_0. We keep the same parameter values, apart from setting $k_0 = 1$.

Running Program 7.10 with these parameters and optimising objective function 1, the incremental gain is found to be equal to $\gamma^2 = 0$. This means that there is a feasible $W_0(x_2, \tilde{x}_2)$ for every non-negative value of γ^2, even when $\gamma^2 = 0$. In this case, by selecting objective OBJ3 for fixed $\gamma^2 = 0$ and minimising $W_0(-R, -R)$ the uniform upper bound

$$\|x_2 - \tilde{x}_2\|^2 \leq W_0(-R, -R) = 0.0838$$

can be constructed. This means that, regardless of the parameter perturbation's effect on the upstream layer, the downstream output will not deviate by more than this amount, given the initial condition $k_0 = 1$. This is true for arbitrarily large values of $\|u_2 - \tilde{u}_2\|^2$, corresponding to extreme perturbations of the upstream system. Indeed, this value of $W_0(-R, -R) = 0.0838$ is an upper bound on each entry of the first column of Table 7.1, corresponding to $k_0 = 1$. However, for the parameters chosen it gives a very conservative bound.

Objective function OBJ3 in Program 7.10 may allow for $W_0(-R, -R)$ to be further optimised. Other fixed values of $\gamma^2 > 0$ may result in better error bounds as the input response $\|u_2 - \tilde{u}_2\|^2$ varies. Figure 7.1 shows the result of taking 11 values of $\gamma_n^2 = 0.006n$ for $n = 0, 1, \ldots, 10$. For each fixed value of γ_n^2, Program 7.10 calculates a feasible $W_{0,n}$ and optimises the value of $W_{0,n}(-R, -R)$ for each γ_n^2. For each $n = 0, 1, \ldots, 10$, the eleven straight lines $\gamma_n^2 \|u - \tilde{u}\|^2 + W_0(-R, -R)$ define valid upper bounds on $\|x_2 - \tilde{x}_2\|^2$, each of which is a tighter bound than the others for different regions of $\|u_2 - \tilde{u}_2\|^2$.

Alternatively, the known bounds on $\|u_2 - \tilde{u}_2\|^2$ given by the first column of Table 7.2 can be used in objective function OBJ2 of Program 7.10. This strategy results in upper bounds for the response $\|x_2 - \tilde{x}_2\|^2$ of 0.0336 and 0.0517 respectively. These bounds are clearly less conservative than 0.0838 found using $\gamma^2 = 0$. However, comparison to the first column in Table 7.1 shows a significant level of conservatism in the incremental gain analysis.

Notwithstanding this conservatism, the results of choosing non-zero γ^2 or making use of the upstream bounds demonstrate that, by incorporating information about the upstream response $\|u_2 - \tilde{u}_2\|^2$ into the downstream SOS program, we can improve on the bounds found by simply calculating the incremental gain. This effect will become even clearer when we use the SSF approach to further generalise Program 7.10.

Moving K from excess to scarcity The final case of the dynamics (7.13) is where $y_0 < k_0 < x_0 + y_0$, so that again $R = \min(x_0 + y_0, k_0) = k_0$. In terms of the original dynamics (7.12), this corresponds to the situation where K is in sufficient concentration to consume the initial concentration of Y, but the production of additional Y from X eventually causes scarcity of K compared to Y. The parameter value corresponding to this case is given by $k_0 = 3$.

For this parameter choice, there is no value of γ^2 for which a feasible $W_0(x_2, \tilde{x}_2)$ can be found to satisfy the SOS condition in Program 7.10. Hence the incremental gain approach fails for

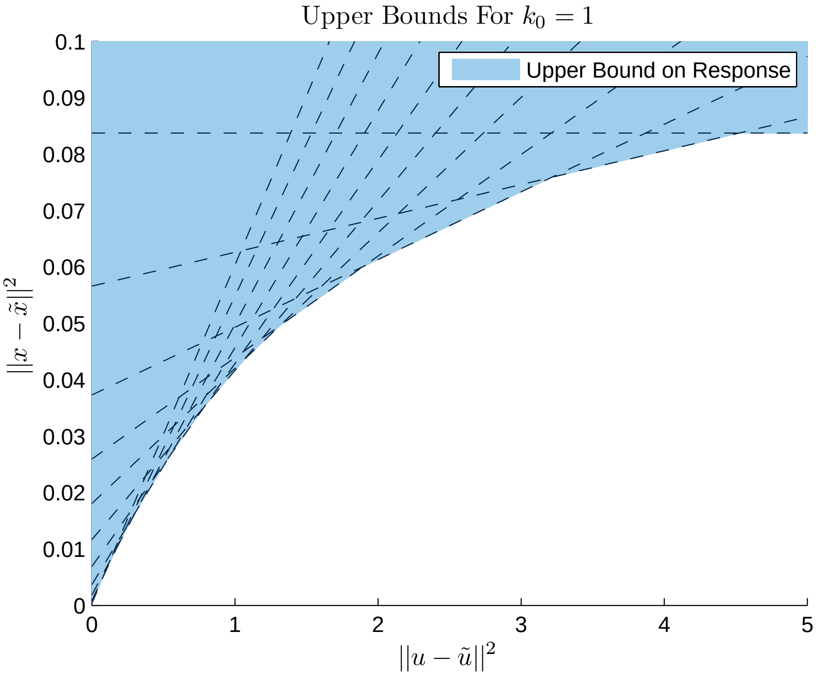


Figure 7.1: The 11 valid upper bounds shown above are constructed by fixing the values of $\gamma_n^2 = 0.006n$ for $n = 0,\ldots,10$ and optimising $W_{0,n}(-R,-R)$ according to Program 7.10. Each straight line defines the bound $\|x_2 - \tilde{x}_2\|^2 \le \gamma_n^2 \|u_2 - \tilde{u}_2\|^2 + W_{0,n}(-R,-R)$.

this parameter choice, and without a more flexible approach a decomposed error bound cannot be found. In the following section, the SOS implementation of incremental gain given by Program 7.10 is replaced by Program 7.5, in order to search for a more general SSF.

7.4.4 Constructing SSFs

Section 7.3.1 showed that the incremental gain framework constructs an SSF of the form (7.6) with $K = 1$ and constant $W_1 = \gamma^2$. We can now apply the increased flexibility of the SSF formulation to the current example, in order to define an SOS program of the form given in Program 7.5 that can be used in place of Program 7.10 to construct an upper bound on $\|x_2 - \tilde{x}_2\|^2$ given the perturbation of β_2 to $\tilde{\beta}_2$.

Program 7.11 Suppose we are given K upstream supply rates $s_k(u_2, \tilde{u}_2)$ for $k = 1,\ldots,K$. Define the $K+1$ SOS polynomial decision variables $W_k(x_2, \tilde{x}_2) \in \Sigma$ for $k = 0,1,\ldots,K$. For the additional variable vector $\boldsymbol{w} = (w_1,\ldots,w_K)^T$, define also the $K+6$ SOS polynomial decision variables $\rho_i(x_2,\tilde{x}_2,u_2,\tilde{u}_2,\boldsymbol{w}) \in \Sigma$, and $\tilde{\rho}_j(x_2,\tilde{x}_2,u_2,\tilde{u}_2,\boldsymbol{w}) \in \Sigma$, for $i, j = 1,\ldots,3$, and $\tau_k(x_2,\tilde{x}_2,u_2,\tilde{u}_2,\boldsymbol{w})$ for $k = 1,\ldots,K$.

Minimise the objective function $W_0(-R,-R)+\sum_{k=1}^{K} E_k^2 W_k(-R,-R)$, where E_k^2 is a pre-determined upper bound on the upstream response $\int_0^\infty -s_k(u_2,\tilde{u}_2)\,\mathrm{d}t \le E_k^2$, subject to the SOS constraint

$$-\left[\dot{W}_0 + \sum_{k=1}^{K}\left(w_k\dot{W}_k + s_k W_k\right) + (x-\tilde{x})^2\right] + \sum_{i=1}^{3}\rho_i\theta_i + \sum_{j=1}^{3}\tilde{\rho}_j\tilde{\theta}_j - \sum_{k=1}^{K}\tau_k w_k \in \Sigma.$$

Here the polynomials $\theta_i(x_2,u_2)$ and $\tilde{\theta}_j(\tilde{x}_2,\tilde{u}_2)$ defining the estimate (7.16) were given in Pro-

Bound on $\|x_2 - \tilde{x}_2\|^2$	$s_2 = -(u_2 + \tilde{u}_2)^2$		$s_2 = -(u_2)^2$	
$k_0 = 3$	$\deg(W_k) = 2$	$\deg(W_k) = 4$	$\deg(W_k) = 2$	$\deg(W_k) = 4$
$\tilde{\beta}_2 = 2$	0.7560	0.7158	0.8275	0.8251
$\tilde{\beta}_2 = 3$	1.3130	1.2193	1.4985	1.4864

Table 7.3: Upper bounds on $\|x_2 - \tilde{x}_2\|^2$, found through SSFs constructed using Program 7.11 with $K = 2$, for two different choices of s_2 in Program 7.9. In each case, $s_1 = (u_2 - \tilde{u}_2)^2$. These errors correspond to the middle column in Table 7.1, where $k_0 = 3$.

gram 7.10 above. ◊

Suppose first that $K = 1$, and the only corresponding upstream supply rate is set to $s_1 = -(u_2 - \tilde{u}_2)^2$. The upper bound $\|u_2 - \tilde{u}_2\|^2 \le E_1^2$ can be found by applying Program 7.9, since $u_2 = z_1$ and $\tilde{u}_2 = \tilde{z}_1$. The two values of E_1^2 for the different parameter perturbations are given by the first column in Table 7.2, and can be substituted into the objective function of Program 7.11 with $K = 1$. For the varying values of $k_0 = 1, 3, 5$, this SOS program (where W_1 is set to be a polynomial of degree 2 or 4) cannot improve upon any of the bounds found through the incremental gain approach of Program 7.10 (where $W_1 = \gamma^2$ is constant). Importantly, it still cannot return any bound when $k_0 = 3$.

However, now suppose that $K = 2$, with $s_1 = -(u_2 - \tilde{u}_2)^2$ as above, and with $s_2 = -(u_2 + \tilde{u}_2)^2$. In Section 7.4.2, two instances of Program 7.9 were used for $k = 1, 2$ to find storage functions V_k certifying the dissipativity of the upstream error system (7.9a) with respect to each of the supply rates s_k. The resulting values of each E_k^2 are given in the kth column of Table 7.2, for each perturbed value $\tilde{\beta}_2$.

For $k_0 = 3$, consider Program 7.11 with $K = 2$, using these supply rates s_k. Unlike the $K = 1$ case discussed above, this program is now feasible, meaning that an upper bound of

$$\|x_2 - \tilde{x}_2\|^2 \le W_0(-R, -R) + \sum_{k=1}^{2} E_k^2 W_k(-R, -R)$$

can be constructed from the resulting SSF. The first two columns of Table 7.3 evaluate the minimal value of this upper bound in the two cases where $\deg(W_k) = 2$ and $\deg(W_k) = 4$, where the objective function of Program 7.11 uses the values of E_k^2 given in Table 7.2.

It is not clear how close this choice of s_2 is to being optimal, in the sense of providing a tight upper bound; other choices of s_2 can return upper bounds. Suppose instead that $s_2 = -(u_2)^2$. A single additional upstream analysis, implemented by using $s_2 = -(u_2)^2$ in Program 7.9, returns an upper bound of $\|u_2\|^2 \le E_2^2 = 8.5419$. The resulting error bounds found using this alternative choice of s_2 are given in the final two columns of Table 7.3. Clearly, the second choice of s_2 gives more conservative bounds than the original choice of $s_2 = -(u_2 + \tilde{u}_2)^2$. However, improvements on the original choice may exist. Clearly, an important open problem for research into the optimal application of the SSF method is to determine the best choices of upstream supply rates s_k to achieve the tightest possible bounds on the downstream response.

Section 7.6 will examine in more detail the reason that setting $K = 2$ and including $s_2 = -(u_2 + \tilde{u}_2)^2$ in the SSF formulation of Program 7.11 improves the outcome of the decomposed analysis past the incremental gain result.

7.4.5 Reducing computational burden

As stated in Section 7.2.5, an important motivation for the decomposition of dissipativity analysis is to speed computation. Each instance of the unstructured dissipativity analysis, carried out in Program 7.8 and used to populate Table 7.1, is an SOS program in 6 state variables, taking an average of 50.5 seconds to complete. However, in the structured case, each instance of the upstream analysis given by Program 7.9 for each s_k is an SOS program in only 4 state variables, taking an average of only 2.95 seconds to complete.

In order for the structured computation to be faster than the unstructured computation, the downstream programs given by Program 7.10 or Program 7.11 should be quicker than the unstructured program. When k_0 is sufficiently small or sufficiently large, Program 7.10 is an SOS program in the 4 state variables $(x_2, \tilde{x}_2, u_2, \tilde{u}_2)^T$, and takes an average of 3.2 seconds to complete (for objective function 2). However, in the case of intermediate values of k_0 where an SSF is required with $K = 2$, Program 7.11 is an SOS program in 6 state variables $(x_2, \tilde{x}_2, u_2, \tilde{u}_2, \boldsymbol{w})^T$, and is therefore not significantly cheaper than the original unstructured computation, taking 46.2 seconds in the case of $\deg(W_k) = 2$, and 48.0 seconds for $\deg(W_k) = 4$.

Note that some of this cost can be offset by the fact that the resulting W_k, for $k = 0, 1, 2$, found by Program 7.11 for pre-determined values of E_k^2, remain valid (although suboptimal) components for constructing SSFs to bound different upstream perturbations. For example, suppose the perturbed parameter value is now $\tilde{\beta} = 4$ rather than 2 or 3 as in Table 7.3. Program 7.11 does not have to be run again in order to bound the dynamic response; the new perturbation to the upstream system only requires the relatively quick upstream computation of Program 7.9 to produce a valid upper bound on the difference in trajectories.

Nevertheless, Section 7.2.5 identified a criterion for the number of variables used in Program 7.4 and Program 7.5 to ensure that each subsystem's dissipativity calculation is cheaper than the unstructured case. In this example, the dimensions $(N_1, N_2, N_{\boldsymbol{u}_2}, K) = (4, 2, 2, 2)$ used in Program 7.9 and Program 7.11 imply that the inequality $N_{\boldsymbol{u}_2} + K < N_1$ does not hold, and thus the downstream SOS program is not significantly cheaper than the unstructured case.

7.5 Example: A Larger Cascade

Section 7.4.5 showed that the computational benefit of using SSFs for dissipativity analysis is obscured by the relative simplicity of the example used up to now. Consider instead the BRN with reactions (3.26), the dynamics of which form an ODE system (3.27) of dimension $\text{rank}(\boldsymbol{S}) = N = 6$. Note that this BRN contains the smaller BRN (7.12) as a layer.

The dynamics of this BRN can be decomposed into the cascade form (7.1), and therefore belongs to the class of systems considered in this chapter. This derivation is given in Appendix C. The upstream system has state $\boldsymbol{x}_1 = (x_{1,1}, x_{1,2}, x_{1,3})^T$ with dynamics

$$\begin{aligned}\begin{bmatrix}\dot{x}_{1,1}\\ \dot{x}_{1,2}\\ \dot{x}_{1,3}\end{bmatrix} &= \begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\alpha_1 x_{1,1}\\ \alpha_2 x_{1,2}\\ \alpha_{12}\left(p_{1,0} - P - x_{1,1} - x_{1,3}\right)\left(p_{2,0} - P - x_{1,2} - x_{1,3}\right)\end{bmatrix}\\ &= \boldsymbol{S}^1 \boldsymbol{v}^1(\boldsymbol{x}_1),\end{aligned} \tag{7.17a}$$

with initial conditions $\boldsymbol{x}_1(0) = (p_{1,0}, p_{2,0}, -P)^T$ and zero steady state. Here $P = \min(p_{1,0}, p_{2,0})$ is a constant parameter depending on the initial concentrations of P_1 and P_2.

The downstream system is driven by the scalar input $u_2 = z_1 = x_{1,3}$ equal to the output of the upstream system. The state $\boldsymbol{x}_2 = (x_{2,1}, x_{2,2}, x_{2,3})^T$ has dynamics

$$\begin{aligned}\begin{bmatrix}\dot{x}_{2,1}\\ \dot{x}_{2,2}\\ \dot{x}_{2,3}\end{bmatrix} &= \begin{bmatrix}1 & -1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\beta_1(-x_{2,2}-x_{2,1})(P+u_2-x_{2,1})-\beta_{-1}x_{2,1}\\ \beta_2 x_{2,1}\\ \beta_3(x_0+y_0-R+x_{2,2}-x_{2,3})(k_0-R-x_{2,3})\end{bmatrix}\\ &= \boldsymbol{S}^2\boldsymbol{v}^2(u_2, \boldsymbol{x}_2) \end{aligned} \tag{7.17b}$$

with initial conditions $\boldsymbol{x}_2(0) = (0, -x_0, -R)$ and zero steady state, where $R = \min(x_0 + y_0, k_0)$. The output of this subsystem is defined to be $z_2 = x_{2,3}$, which is assumed to be equal to the output $y = z_2$ of the combined system (7.17).

7.5.1 Parametric perturbations upstream: Two layers

Suppose that the parameter α_{12} is perturbed, and we aim to find a bound on the response of the output $\|y - \tilde{y}\|^2$, where the notation $\tilde{\cdot}$ again denotes the perturbed system. In this case, the unstructured error estimation computation given by Program 3.3 will require an unstructured SOS program in 12 variables $(\boldsymbol{x}_1, \boldsymbol{x}_2, \tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2)$, which cannot be solved in any reasonable timeframe on a standard desktop computer.

This example is intended to demonstrate that exploiting the cascade form of (7.17) to calculate an SSF provides significant computational benefits. A brief dimensional analysis supports this assertion. The error system associated with the upstream layer's dynamics in (7.17a) has dimension $N_1 = 6$, corresponding to the variables $(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_1)$. Thus, for given upstream storage functions s_k, with $k = 1, \dots, K$, each instance of the upstream dissipativity computation in Program 7.4 is an SOS program in 6 state variables, which will clearly be far faster than an SOS program in 12 variables. The error system associated with the downstream dynamics (7.17b) has state dimension $N_2 = 6$, and input dimension $N_{\boldsymbol{u}_2} = 2$. Thus the total number of states used in Program 7.5 will be $N_2 + N_{\boldsymbol{u}_2} + K = 8 + K$. Therefore, we expect that its implementation will be significantly cheaper than that of the unstructured Program 3.3 for smaller values of $1 \le K < 4$.

Suppose that the nominal parameter $\alpha_{12} = 1$ is perturbed to $\tilde{\alpha}_{12} = 2$. The other parameters remain constant, given in Table 7.4. Substituting the error system corresponding to the upstream subsystem (7.17a) into Program 7.4 for $s_1(z_1, \tilde{z}_1) = -(z_1 - \tilde{z}_1)^2$ results in an upper bound $\|z_1 - \tilde{z}_1\|^2 \le 0.0405$ in approximately 75 seconds. Upon attempting to use this form of s_1 in Program 7.5, with $K = 1$, we find that there is no feasible pair W_0, W_1 of SOS polynomials that can be used to construct an SSF for the downstream output in this parameter range. As in the previous example, a parallel upstream dissipativity computation, putting $s_2 = -(z_1 + \tilde{z}_1)^2$ into another instance of Program 7.4, results in an upper bound $\|z_1 + \tilde{z}_1\|^2 \le 4.609$ in a similar time of approximately 75 seconds. Both s_1 and s_2 may now be substituted into Program 7.5 to attempt to construct an SSF for $K = 2$. This SOS program is in $N_2 + N_{\boldsymbol{u}_2} + K = 6 + 2 + 2 = 10$ variables, which is clearly an improvement on the unstructured program's $N = 12$ variables. By minimising the value of $W_0 + 0.0405W_1 + 4.609W_2$, an upper bound of $\|y - \tilde{y}\|^2 \le 0.6717$ can be found in just over 2 hours, searching over $W_k \in \Sigma$ of degree 4.

α_i	α_{12}	$\tilde{\alpha}_{12}$	β_i	$p_{0,1}$	$p_{0,2}$	x_0	y_0	k_0
1	1	2	1	2	1	2	2	5

Table 7.4: Parameter values used in computation of error. Nominal parameter α_{12} is perturbed to $\tilde{\alpha}_{12}$.

The final SOS program is clearly an expensive calculation. However, as noted in the previous example, the resulting polynomials W_0, W_1 and W_2 can now be recycled to bound the response $\|y-\tilde{y}\|^2$ to a different perturbation of the upstream system, albeit more conservatively. Only two parallel instances of the much cheaper Program 7.4 are required to find new upstream storage functions V_1 and V_2, which are substituted into $V = W_0 + W_1 V_1 + W_2 V_2$ to give a new, valid SSF in a timescale on the order of one minute, rather than hours.

7.5.2 Parametric perturbations upstream: Three layers

Alternatively, the results of the dissipativity analysis in Section 7.4 can now be applied to this larger example. This is implemented by the further decomposition of the downstream dynamics (7.17b) into the form similar to (7.13). The resulting dynamics are

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{S}^1 \boldsymbol{v}^1(\boldsymbol{x}_1) \text{ as in (7.17a)}, \tag{7.18a}$$

$$\dot{\boldsymbol{x}}_2 = \begin{bmatrix} \dot{x}_{2,1} \\ \dot{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1(-x_{2,2} - x_{2,1})(P + u_2 - x_{2,1}) \\ \beta_2 x_{2,1} \end{bmatrix}, \tag{7.18b}$$

$$\dot{x}_3 = \beta_3(x_0 + y_0 - R + u_3 - x_3)(k_0 - R - x_3), \tag{7.18c}$$

with the cascade implied by $u_2 = z_1 = x_{1,3}$ driving the second subsystem, and $u_3 = z_2 = x_{2,2}$ driving the third subsystem. The initial conditions of the nominal and perturbed systems are given by $\boldsymbol{x}_1(0) = \tilde{\boldsymbol{x}}_2(0) = (p_{1,0}, p_{2,0}, -P)^T$ for layer 1, $\boldsymbol{x}_2(0) = \tilde{\boldsymbol{x}}_2(0) = (0, -x_0)^T$ for layer 2, and $x_3(0) = \tilde{x}_3(0) = -R$ for the final layer.

Observe that the downstream subsystem (7.18c) is of exactly the same form as the downstream subsystem (7.13b). Since the parameters satisfy $k_0 = 5 > x_0 + y_0$, the previous calculation of the incremental gain of (7.13b) as $\gamma^2 = 1$ implies that

$$\|y - \tilde{y}\|^2 = \|x_3 - \tilde{x}_3\|^2 \leq \|u_3 - \tilde{u}_3\|^2 = \|x_{2,2} - \tilde{x}_{2,2}\|^2.$$

This additional decomposition step, coupled with the analysis in the previous example, implies that an SOS program bounding the response $\|x_{2,2} - \tilde{x}_{2,2}\|^2$ of the midstream layer (7.18b) to the perturbation of its input from u_2 to $\tilde{u}_2$ is sufficient to bound $\|y - \tilde{y}\|^2$.

We consider again the SSF approach with $K = 2$. The two instances of Program 7.4 corresponding to $s_1 = -(z_1 - \tilde{z}_1)^2$ and $s_2 = -(z_1 + \tilde{z}_1)^2$ have already been carried out in Section 7.5.1, resulting in upstream storage functions V_1 and V_2 such that

$$V_1\big(\boldsymbol{x}_1(0), \tilde{\boldsymbol{x}}_1(0)\big) = E_1^2 = 0.0405,$$
$$V_2\big(\boldsymbol{x}_1(0), \tilde{\boldsymbol{x}}_1(0)\big) = E_2^2 = 4.609.$$

The downstream dissipativity calculation is implemented by substituting the error system of (7.18b) and supply rates $s_k(u_2, \tilde{u}_2)$, for $k = 1, 2$, into Program 7.5. This SOS program is in 8 variables $(x_{2,1}, x_{2,2}, \tilde{x}_{2,1}, \tilde{x}_{2,2}, u_2, \tilde{u}_2, w_1, w_2)$, which is clearly a more significant improvement over 12 variables than 10 was in the two-layer case of Section 7.5.1. The upper bound on $\|x_{2,2} - \tilde{x}_{2,2}\|^2$ found by minimising $W_0 + W_1 E_1^2 + W_2 E_2^2$ (evaluated at the initial conditions) implies that

$$\|y - \tilde{y}\|^2 \leq \|x_{2,2} - \tilde{x}_{2,2}\|^2 \leq 0.7003,$$

which can be achieved in around 12 minutes. This compares with the simulated responses of $\|x_{2,2} - \tilde{x}_{2,2}\|^2 \approx 0.0515$, and $\|y - \tilde{y}\|^2 \approx 0.0479$. Again, the conservatism of each of these upper bounds is the cost of the SSF approach, which allows an SOS-derived bound in a reasonable time frame. Furthermore, it is likely that different choices of upstream s_i could produce less conservative bounds.

7.6 Discussion

This chapter has introduced the novel framework of Structured Storage Functions (SSFs). This framework allows system structure (specifically, cascade structures) to be exploited in order to decompose the SOS implementation of dissipativity analysis introduced in Chapter 2. We showed in Section 7.3 that this new framework can be applied to estimating the response of large-scale polynomial systems to structural or parametric perturbations. In particular, the examples in Section 7.4 and Section 7.5 illustrated this technique for decomposed parameter sensitivity analysis.

Clearly, another application of this technique would be to consider the propagation of the error incurred by the structured reduction of a system, by reducing subsystems. For example, rather than perturbing a parameter in the enzyme reaction (7.13a), it could be reduced through one of the QSS methods discussed in Section 3.2. The response in the output of the upstream system can then propagate to perturb the output y of the downstream system to $\tilde{y}$. Note that, similarly to the parameter perturbation case, when reducing the upstream layer the downstream polynomials $W_k \in \Sigma$ do not need to be re-calculated to provide a valid upper bound on $\|y - \tilde{y}\|^2$. In addition to applications in comparing perturbed system trajectories, there are also many possible applications of the SSF approach to questions of the controllability, observability, passivity (and so on) of structured, layered systems, since each of these properties can be expressed in terms of dissipativity [60].

In this chapter, the SSF concept is limited to cascade systems. The linear combination dissipativity technique briefly discussed in Section 7.2.1 is easily extended to more general interconnections of subsystems. To ensure that the SSF strategy is as widely applicable as the linear combination approach, the results of this chapter must be extended to consider the decomposed dissipativity analysis of subsystems in feedback. Suppose each 'upstream' Program 7.4 is calculated for the supply rates $s_k(\boldsymbol{u}_1, \boldsymbol{z}_1)$, now with an input $\boldsymbol{u}_1 = \boldsymbol{z}_2$ equal to the 'downstream' output. If V_k can be calculated for these s_k, then these may be used to construct an SSF through an analogy of Program 7.5. This approach will form the starting point of future research on this methodology, which may also be informed by classical systems theory concepts such as small gain and structured uncertainty [217].

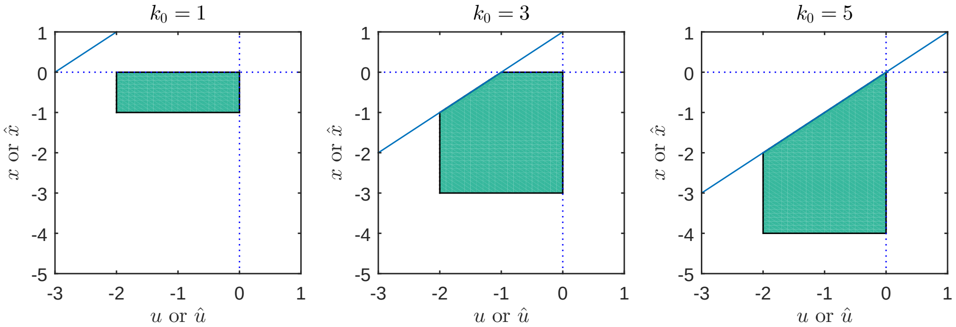


Figure 7.2: The physically relevant state space and input space $\mathscr{D}_2 \times \mathscr{U}_2$ for $x_0 = y_0 = 2$ and three representative values of k_0; the inequalities are all implied by the requirement for each of the individual species concentrations to be non-negative.

7.6.1 Choosing upstream storage functions

Consider again the simple incremental gain-type SSF applied to the example described in Section 7.4.3. While this approach works for values of k_0 sufficiently large and sufficiently small, in the intermediate region no constant could be found such that Program 7.10 is feasible. Figure 7.2 depicts the domain $\mathscr{D}_2 \times \mathscr{U}_2$ of the downstream layer's state x and input u for three representative values of k_0. In the formulation of Program 7.10, the input trajectories are constrained such that the values $u(t)$ and $\tilde{u}(t)$ at each time $t \geq 0$ are in these regions.

However, the SSF formulation imposes an additional, implicit constraint on the trajectories of the inputs to the downstream system. Suppose W_0 and γ^2 have been found by Program 7.10. Then dissipativity has been shown for all inputs u_2 and $\tilde{u}_2$ such that a time-dependent variable $v(t) \geq 0$ exists with $\int_0^t (u_2 - \tilde{u}_2)^2 \, \mathrm{d}t \leq v(0) - v(t) < \infty$. Thus this formulation shows dissipativity of the downstream system for inputs such that $u_2 - \tilde{u}_2 \to 0$ sufficiently quickly. However, it is not necessarily the case that the inputs u_2 and $\tilde{u}_2$ approach 0 as $t \to \infty$.

One conjecture suggested by this example, and in particular the failure of incremental gain approaches for only the case $k_0 = 3$, is that the discontinuity in the gradient of the steady state of x_2 as a function of constant u_2 means that dissipativity cannot be proven. If, however, the space of input trajectories is further constrained such that $u_2(t), \tilde{u}_2(t) \to 0$ as $t \to \infty$, or even simply $u_2(t), \tilde{u}_2(t) > -1$ for $t > T$, then dissipativity should be provable. This was the case in Section 7.4.4, where both choices of s_2 imply that the space of input trajectories is further constrained such that $(u_2 + \tilde{u}_2)^2 \to 0$, or $(u_2)^2 \to 0$. In other words, the downstream system's dissipativity was certified through an SSF that 'imports' additional information about the two input trajectories. This interpretation appears to link the SSF formulation to the ideas underlying IQCs [127, 181, 198]. This connection will be interesting to explore further in any future developments.

The conjecture above may also relate to the results on steady-state signal propagation in Chapter 6. For example, the derivative $\mathrm{d}\bar{\boldsymbol{x}}^2/\mathrm{d}\bar{\boldsymbol{x}}^1$ describing how the layer 2 steady state varies with the layer 1 steady state will not exist for $k_0 = 3$ if $x_{1,2} = u_2 \to -1$ at steady state. Hence, another possible direction of future research into the SSF formulation will be to relate the analytical results of Chapter 6 to the computational techniques of this chapter, potentially to inform the optimal number K and choices s_k of upstream supply rates.

Chapter 8

Conclusions

The focus of this thesis has been to provide strategies for the analysis of high-dimensional ODE models of Biomolecular Reaction Networks (BRNs). Specifically, the techniques investigated in this work are intended to accomplish the complementary aims of model reduction and model decomposition. The main conclusions of the thesis are summarised below, after which we briefly outline some future directions for prospective further research.

8.1 Summary

Chapter 2 reviewed previous work in order to introduce the details of the systems analysis methods used in this thesis. In this chapter we showed how Sum of Squares (SOS) programming is applied to certify the dissipativity of autonomous nonlinear polynomial systems. The first contribution of this thesis, in Chapter 3, was to apply this technique to the comparison of two different ODE models. The comparison, or error, is defined as the $\mathscr{L}_2$ norm of the difference in outputs; by the dissipativity-based approach, the worst-case error over a set of initial conditions can be estimated. Providing a worst-case upper bound with a single SOS program can be significantly more efficient than carrying out many simulations of large-scale systems across all initial conditions, and provides guaranteed upper bounds that simulation would only be able to approximate. The two examples in Chapter 3 demonstrated that this technique has applications to estimating the error incurred by model order reduction, and also to quantifying the sensitivity of a system's output to parametric perturbations. A report of the model reduction application has been published in [150].

In common with many other applications of SOS programming, a key limitation on the techniques in Chapter 2 and Chapter 3 is the rapid increase in computational resources required as the system's dimension and its degree of nonlinearity increase. This has a particularly important effect on the model reduction application of Chapter 3. As indicated by the enzyme kinetics example in Section 3.2 (and the definition of $\boldsymbol{M}^f$ in (5.7), for the general case), quasi-steady state (QSS) methods for model reduction can often produce rational polynomial vector fields $\tilde{\boldsymbol{f}}$ or output maps $\tilde{\boldsymbol{h}}$. This requires dissipativity inequalities of the form (2.7c), where multiplication by a common denominator results in the search for SOS polynomials of relatively high degree. Furthermore, the number of variables in any SOS program bounding the error between two models is equal to the summed dimensions of each model. Unfortunately, model reduction is usually motivated by a high-dimensional nominal system. Thus, regardless of how much simpler the reduced system is, the SOS program comparing the two outputs is likely to have a high computational cost, and hence will be impractical to carry out directly.

To approach the task of scaling up the procedures introduced in Chapter 3 we then considered the complementary strategy of model decomposition. Chapter 4 has introduced a novel framework for the decomposition of large-scale BRNs, called *layered* decomposition. The key distinction between the new framework and the established approach of modular decomposition arises from the observation that the state of each module corresponds to a subset of the species. In contrast, the state of each layer takes values in the original state space, so that multiple layers may contribute to the dynamics of any species. The layered states' values are large-dimensional vectors; we showed, however, that layers are subsystems of the original system when these vectors are embedded in a smaller-dimensional subspace of the original state space.

Once the layered decomposition framework is defined, the fundamental question of how to choose this decomposition remains. The remainder of Chapter 4 and the subsequent two chapters propose three potential answers to this question. The first answer would be to ensure that if the large-scale system is stable, its expression as an interconnection of layers is also stable. The second half of Chapter 4 showed that this condition is implied by ensuring that the total dimension of the layers is as small as possible. For a decomposition strategy based on decomposing the reaction set, we then showed how to define layers that satisfy this condition.

A second, more biologically-informed, perspective on the question of choosing a decomposition is given in Chapter 5. The basis of this strategy is that the reactions of a given network are often separated by timescale. This separation implies a natural partition of the reactions, and hence a natural layered decomposition. In work which has also been published in [151, 152], Chapter 5 described the application of layering by timescale to the quasi-steady state approximation technique. Thus, reduced models can be produced as a consequence of a layered decomposition strategy, further illustrating the close connection between model decomposition and model reduction.

The third response to the question 'how to layer' considers the case where timescale separation is no longer apparent. The approach taken in Chapter 6 was to consider how well a layered decomposition partitions the large-scale system. This concept was quantified in terms of the way in which signals propagate between layers. In Chapter 6, for each given layered decomposition we derived (through a decomposition of steady-state parameter sensitivity analysis) an associated inter-layer communication structure. The relative quality of each possible layered decomposition was then identified with the sparsity of its associated communication structure. We suggest that since this criterion provides a means for choosing between two layered decompositions, it can form the basis of future algorithmic methods for automatically detecting layered architectures in large-scale systems.

The final chapter of the thesis returned to the problem identified at the end of Chapter 3: the computational challenge of applying SOS techniques to compare BRN models, at least one of which is assumed to be of large dimension. This chapter showed a further application of decomposition techniques, in the special case of cascade systems, to the certification of dissipativity. Chapter 7 defined Structured Storage Functions (SSFs), formulated to reflect the underlying cascade structure of the system [154]. As a result, the SOS program implementing the search for a

SSF can be distributed into multiple independent SOS programs that are of significantly smaller dimension than the unstructured method used in Chapter 2. This chapter then applied the SSF formulation to the output comparison problems introduced in Chapter 3. We showed that incremental gain, which is a previously studied approach to characterising how perturbations to a system's input propagate to its output, is special case of SSF. However, the additional flexibility of SSFs beyond the incremental gain framework allows for a larger range of output comparison SOS programs to be successfully decomposed. As a consequence, we can significantly reduce the computational burden of these tasks, enabling error bounds to be calculated when unstructured methods fail.

8.2 Outlook

The work in this thesis, summarised above, has also formed the basis of a number of additional research projects which have not been included in this thesis. In particular, references [148, 149] are two further papers currently under review. In [149] we adapt the layered formulation (4.7) to enforce the dynamic interconnection structure into a cascade. This allows each layer's dynamics to be interpreted as a simulation of the incremental effect of adding a set of reactions to a given network. This can be used to systematically quantify the pairwise interactions between layers, which vary depending on the other layers present. The model-comparison techniques of Chapter 2 and Chapter 3 are applied in [148] to compare the dynamics of loaded and unloaded modules, thereby quantifying the worst-case effect of retroactivity.

One key assumption underlying the entire thesis is that the concentration vector $\boldsymbol{x}(t) \to \bar{\boldsymbol{x}}$ approaches steady state $\bar{\boldsymbol{x}}$ as $t \to \infty$. In order to extend the conclusions and techniques developed here to a wider range of BRNs, the next stage is to consider other types of systems, such as those that oscillate along limit cycles. In particular, using the $\mathscr{L}_2$ norm as the model comparison measure $\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|$ is no longer likely to be applicable, since even if both systems approach the same limit cycle there is likely to be a phase lag such that the $\mathscr{L}_2$ norm is infinite. It will be interesting to determine a different polynomial storage function that can be used to estimate a meaningful measure of the output difference.

Furthermore, introducing quantitative measures of the quality of a particular model reduction allows two competing candidate reductions to be directly compared, and one selected. Thus, an important potential application of this technique is for the design of a model reduction algorithm. The strategy adapted from [8] and briefly described in Section 3.4.1 is to sequentially generate a set of candidate reduced models, and at each step choose the reduction with minimal worst-case error. In that discussion, the candidate reductions were simple QSS approximations of single states. There is much more work to be done on how to generate the candidate reductions that can be compared. For example, the layered approach to QSS approximation given in Chapter 5 provides another strategy for generating potential reductions, based on putting layers at quasi-steady state. Furthermore, approximation strategies for BRNs other than QSS approximations should also be investigated.

Recall that the layered decomposition framework defines subsystems with input–output dynamics given by (4.8). These dynamics remain the same whether or not the layer is integrated in a larger system; the presence of the context only changes the input. This framework will have significant applications in the context of Synthetic Biology to the important challenge of constructing synthetic BRNs as the interconnection of designed components. As with Systems Biology, this challenge has up to now been approached from a modular perspective. However, the key benefit of layered Synthetic Biology will be the definition of subsystems with invariant input–output behaviours, which may therefore provide more reliable definitions of synthetic components. For example, we should be able to construct novel control layers that directly affect the inputs $\boldsymbol{u}^l$ of each of the other layers.

In addition to the steady-state (and potentially dynamic) parameter sensitivity techniques described in Chapter 6, there may be other analytical approaches to inter-layer communication. Two important, related system properties are the controllability and observability of a system. Rather than deriving an interlayer communication map from the layered decomposition of parameter sensitivity analysis, a layered decomposition of these alternative system properties may provide alternative definitions of interlayer communication. Alternatively, the exploration of signal propagation may be extended to consider how noise propagates between layers; a good layering may be one in which local stochastic effects within the subsystem are not globally amplified, or remain relatively uncorrelated between subsystems [33]. These alternative approaches may provide different criteria for the quality of any particular layered decomposition.

Section 6.5.1 briefly explored how the interlayer communication structure can be used to detect layered structures. Similarly to the model reduction algorithm in Section 3.4.1, a layering algorithm also requires us to generate layered decompositions to compare. The brute-force approach, based on the combinatorial selection of all possible partitions of the reactions, is highly impractical for large-scale systems. In the related field of community detection in static networks, the problem of community decomposition (optimally partitioning a node-set) was approached with heuristic algorithms based on random walks and the linear algebraic properties of the network's adjacency matrix [25, 145]. Ideally, a solution to the layered decomposition problem may be approached with similar heuristic tools.

Finally, there are many developments of Structured Storage Functions (SSFs) that should improve their applicability to proving the dissipativity of a much larger range of systems. In particular, recall that the SSF theory as described in Chapter 7 assumes that the large-scale system is a cascade of subsystems. As described in Section 7.6, this theory should also be extended to systems that include feedback loops between layers. Furthermore, we discussed in Section 7.6.1 that more research must be done to identify which upstream supply rates should be used for the SSF construction. In other words, we need to choose the supply rates of the upstream subsystem that maximise the amount of information given about the inputs of the downstream subsystem.

If SSFs can be extended to feedback pairs, then they will be particularly useful in the development of the work of Chapter 5 on layered approaches to singular perturbation. Under the QSS approximation, a dynamic layer is approximated to be static, while the other layers are left unperturbed. The SSF approach may allow us to understand how the error of approximating a dynamic layer as a static subsystem propagates in feedback to the other dynamic layers. This may produce a bound on the output error that can be expressed in terms of the system parameters; in particular, as a function of the small parameter μ.

This thesis has demonstrated that model decomposition and model order reduction are both important, closely-related approaches to the analysis of large-scale systems. As the models of evolved and synthetic biomolecular reaction networks grow in scale, these techniques will become increasingly vital to the understanding and design of such systems. Layered approaches to large-scale systems may have a number of applications outside of BRNs. In particular, it may be interesting to investigate connections with other types of model, such as: multi-agent systems with highly-structured communication architectures [64, 204]; cyber-physical systems, such as smart grids of mutually interacting power and communication networks [35]; and multi-modal transport networks [69], amongst others. Furthermore, the language of 'layering as optimisation decomposition' in communication networks [47] suggests important potential developments in the study of biomolecular networks as complex interconnections of coupled optimisation algorithms that operate under biochemical, evolutionary, and ultimately *designed* constraints.

Appendix A

Example SOS Code

The following code is the implementation of Program 3.6, estimating the error in the sQSSA reduction of the enzyme kinetics network in Section 3.2.

```
function [errBoundsol,Vsol] = Enzyme_sQSSA_FixedICs(par,degV)
% To return an upper bound on the error incurred by the sQSSA,
% given system parameters in par.

%% Define dynamics in error system
syms x c tx real
xe = [x;c;tx];

Km = (par.km1 + par.k2)/par.k1;
v = [par.k1*x*(par.E0-c);
    par.km1*c;
    par.k2*c];
dx = [-1 1 0]*v;
dc = [1 -1 -1]*v;
dtx = -par.k2*par.E0*tx/(tx+Km);

dxe = [dx;dc;dtx];

%% Define estimate of state space
phi1 = x*(x+c-par.X0);
phi2 = c*(c-par.E0);
phi3 = tx*(tx-par.X0);

%% SOS program
prog = sosprogram(xe);

% Initialise decision variables
[prog,V] = sossosvar(prog,monomials(xe,1:degV/2));
[prog,sig1] = sossosvar(prog,monomials(xe,0:1));
[prog,sig2] = sossosvar(prog,monomials(xe,0:1));
[prog,sig3] = sossosvar(prog,monomials(xe,0:1));

% Enforce dissipativity condition
psi = Km+tx;
yeTye = (x-tx)^2;
```

```
Vdot = jacobian(V,xe)*dxe;
ineq = -psi*(yeTye + Vdot) + sig1*phi1 + sig2*phi2 + sig3*phi3;
prog = sosineq(prog,ineq);

% Set objective: minimise V evaluated at fixed initial condition
xe0 = [par.X0;0;par.X0];
errBound = subs(V,xe,xe0);
prog = sossetobj(prog,errBound);

% Solve SOS program and return errBoundsol and Vsol
prog = sossolve(prog,2);
Vsol = sosgetsol(prog,V);
errBoundsol = sosgetsol(prog,errBound);
```

Appendix B

Multiple Timescales

This appendix provides the proofs of Proposition 5.6 and Proposition 5.8.

B.1 Layered QSS approximation

Restate Proposition 5.6. *The layered dynamics* (5.13) *at the slowest timescale can be approximated by the ODE system*

$$\dot{\tilde{\boldsymbol{x}}}^1(t) = \boldsymbol{S}^1\bar{\boldsymbol{v}}\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1(t) + \tilde{\boldsymbol{x}}^2(t) + \cdots + \tilde{\boldsymbol{x}}^L(t)\big), \tag{B.1a}$$

$$\tilde{\boldsymbol{x}}^l(t) = \boldsymbol{\phi}^l\big(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1(t) + \cdots + \tilde{\boldsymbol{x}}^{l-1}(t)\big),\ \textit{for}\ l = 2,\ldots,L, \tag{B.1b}$$

if the matrix $\boldsymbol{C}^l(\bar{\boldsymbol{v}}^l)'\boldsymbol{U}^l$ is invertible, for $l = 2,\ldots,L$.

Here, the static nonlinear functions $\boldsymbol{\phi}^l(\boldsymbol{z})$, for $l = 2,\ldots,L$, satisfy the algebraic equations defining the QSS manifolds $\boldsymbol{0} = \boldsymbol{S}^l\bar{\boldsymbol{v}}^l\big(\boldsymbol{z} + \boldsymbol{\phi}^l(\boldsymbol{z})\big)$, and the functions $\bar{\boldsymbol{v}}^l$ are defined recursively as

$$\bar{\boldsymbol{v}}^L(\boldsymbol{z}) = \bar{\boldsymbol{v}}(\boldsymbol{z}), \quad \bar{\boldsymbol{v}}^{l-1}(\boldsymbol{z}) = \bar{\boldsymbol{v}}^l\big(\boldsymbol{z} + \boldsymbol{\phi}^l(\boldsymbol{z})\big) \tag{B.2}$$

for decreasing $l = L, (L-1),\ldots,2$.

Proof. This result is shown by a sequence of QSS approximations at successively slower timescales. Consider first the approximation of the fastest timescale, corresponding to layer L in the layered decomposition (5.13). Since this is an interim approximation, we will denote the states of the approximated layers as $\tilde{\boldsymbol{x}}^l_1$, where the subscript indexes the total number of layers that have been approximated.

Let $\mu \to 0$ at such a rate that we can approximate $(\mu)^{L-1} \approx 0$ such that $(\mu)^{L-2} \gg (\mu)^{L-1}$ is non-zero, to give

$$(\mu)^{l-1}\dot{\tilde{\boldsymbol{x}}}^l_1 = \boldsymbol{S}^l\bar{\boldsymbol{v}}^L\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-1}\tilde{\boldsymbol{x}}^k_1 + \tilde{\boldsymbol{x}}^L_1\Big)\ \text{for}\ l = 1,\ldots,L-1, \tag{B.3a}$$

$$\boldsymbol{0} = \boldsymbol{S}^L\bar{\boldsymbol{v}}^L\left(\boldsymbol{x}_0 + \sum_{k=1}^{L-1}\tilde{\boldsymbol{x}}^k_1 + \tilde{\boldsymbol{x}}^L_1\right). \tag{B.3b}$$

Assuming that $\boldsymbol{C}^L(\bar{\boldsymbol{v}}^L)'\boldsymbol{U}^L$ is an invertible matrix, the IFT can be applied (as in Section 5.2). This theorem implies that the manifold (B.3b) defines a function, expressing the fastest layer's state in terms of the slower layers' states, such that

$$\tilde{\boldsymbol{x}}_1^L = \boldsymbol{\phi}^L\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-1} \tilde{\boldsymbol{x}}_1^k\Big). \tag{B.3c}$$

Substituting (B.3c) into (B.3a), Tikhonov's Theorem implies that the trajectory of the first interim approximation

$$(\mu)^{l-1}\dot{\tilde{\boldsymbol{x}}}_1^l = \boldsymbol{S}^l\,\bar{\boldsymbol{v}}^{L-1}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-1} \tilde{\boldsymbol{x}}_1^k\Big) \text{ for } l = 1,\dots,(L-1), \tag{B.4a}$$

$$\tilde{\boldsymbol{x}}_1^L = \boldsymbol{\phi}^L\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-1} \tilde{\boldsymbol{x}}_1^k\Big), \tag{B.4b}$$

is arbitrarily close to the trajectory of (5.13) as $(\mu)^{L-1} \to 0$, for $\bar{\boldsymbol{v}}^{L-1}(\boldsymbol{z}) = \bar{\boldsymbol{v}}^L(\boldsymbol{z} + \boldsymbol{\phi}^L(\boldsymbol{z}))$.

For a fixed integer $r < L$, suppose that the fastest r timescales have been approximated, such that the slowest $L-r$ layers are dynamic and the fastest r layers are static. Assume that the dynamics of the approximated layers' states $\tilde{\boldsymbol{x}}_r^l$ are given by

$$(\mu)^{l-1}\dot{\tilde{\boldsymbol{x}}}_r^l = \boldsymbol{S}^l\,\bar{\boldsymbol{v}}^{L-r}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-r} \tilde{\boldsymbol{x}}_r^k\Big) \text{ for } l = 1,\dots,(L-r), \tag{B.5a}$$

$$\tilde{\boldsymbol{x}}_r^l = \boldsymbol{\phi}^l\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l-1} \tilde{\boldsymbol{x}}_r^k\Big) \text{ for } l = (L-r+1),\dots,L. \tag{B.5b}$$

We now intend to show by induction that the states $\tilde{\boldsymbol{x}}_{r+1}^l$ of the next interim approximation, defined by assuming that the next fastest dynamic layer $L-r$ is at QSS, have dynamics of the same form as (B.5).

Let $(\mu)^{L-(r+1)} \to 0$ in (B.5), such that $(\mu)^{L-(r+2)} \gg (\mu)^{L-(r+1)}$ does not approach zero. The next interim approximation is given by

$$(\mu)^{l-1}\dot{\tilde{\boldsymbol{x}}}_{r+1}^l = \boldsymbol{S}^l\,\bar{\boldsymbol{v}}^{L-r}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-(r+1)} \tilde{\boldsymbol{x}}_{r+1}^k + \tilde{\boldsymbol{x}}_{r+1}^{L-r}\Big) \text{ for } l = 1,\dots,(L-(r+1)) \tag{B.6a}$$

$$\boldsymbol{0} = \boldsymbol{S}^{L-r}\,\bar{\boldsymbol{v}}^{L-r}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-(r+1)} \tilde{\boldsymbol{x}}_{r+1}^k + \tilde{\boldsymbol{x}}_{r+1}^{L-r}\Big) \tag{B.6b}$$

$$\tilde{\boldsymbol{x}}_{r+1}^l = \boldsymbol{\phi}^l\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l-1} \tilde{\boldsymbol{x}}_{r+1}^k\Big) \text{ for } l = (L-r+1),\dots,L. \tag{B.6c}$$

Under the assumption that $\boldsymbol{C}^{L-r}(\bar{\boldsymbol{v}}^{L-r})'\boldsymbol{U}^{L-r}$ is an invertible matrix, the manifold (B.6b) defines the implicit function expressing the approximation of layer $L-r$ in terms of slower layers as

$$\tilde{\boldsymbol{x}}_{r+1}^{L-r} = \boldsymbol{\phi}^{L-r}\left(\boldsymbol{x}_0 + \sum_{k=1}^{L-(r+1)} \tilde{\boldsymbol{x}}_r^k\right). \tag{B.6d}$$

Substituting this implicit function (B.6d) into the approximated dynamics (B.6a) of the slower layers $l = 1, \dots, (L - r - 1)$ gives the system

$$(\mu)^{l-1}\dot{\tilde{\boldsymbol{x}}}^l_{r+1} = \boldsymbol{S}^l \bar{\boldsymbol{v}}^{L-(r+1)}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{L-(r+1)} \tilde{\boldsymbol{x}}^k_{r+1}\Big) \text{ for } l = 1, \dots, (L - (r+1)), \tag{B.7a}$$

$$\tilde{\boldsymbol{x}}^l_{r+1} = \boldsymbol{\phi}^l\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l-1} \tilde{\boldsymbol{x}}^k_{r+1}\Big) \text{ for } l = (L - r), \dots, L, \tag{B.7b}$$

matching the form of the previous step (B.5). Hence these dynamics are those of the interim approximation where the fastest $(r+1)$ timescales have now been approximated.

By induction, it follows that the sequential approximation of the fastest $L-1$ timescales results in the dynamics

$$\dot{\tilde{\boldsymbol{x}}}^1_{L-1} = \boldsymbol{S}^1 \bar{\boldsymbol{v}}^1(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1_{L-1}), \tag{B.8a}$$

$$\tilde{\boldsymbol{x}}^l_{L-1} = \boldsymbol{\phi}^l\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l-1} \tilde{\boldsymbol{x}}^k_{L-1}\Big) \text{ for } l = 2, \dots, L. \tag{B.8b}$$

Writing the approximated states $\tilde{\boldsymbol{x}}^l = \tilde{\boldsymbol{x}}^l_{L-1}$, it remains to show that $\bar{\boldsymbol{v}}^1(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1) = \bar{\boldsymbol{v}}\Big(\boldsymbol{x}_0 + \sum_{l=1}^L \tilde{\boldsymbol{x}}^l\Big)$, which would imply that the dynamics (B.8) are equal to (5.14). The recursive definitions of $\bar{\boldsymbol{v}}^l$ and $\tilde{\boldsymbol{x}}^l$ imply the inductive relationship

$$\bar{\boldsymbol{v}}^l\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l} \tilde{\boldsymbol{x}}^k\Big) = \bar{\boldsymbol{v}}^{l+1}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l} \tilde{\boldsymbol{x}}^k + \boldsymbol{\phi}^{l+1}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l} \tilde{\boldsymbol{x}}^k\Big)\Big) = \bar{\boldsymbol{v}}^{l+1}\Big(\boldsymbol{x}_0 + \sum_{k=1}^{l+1} \tilde{\boldsymbol{x}}^k\Big),$$

from which it follows that $\bar{\boldsymbol{v}}^1(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1) = \bar{\boldsymbol{v}}(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1 + \dots + \tilde{\boldsymbol{x}}^L)$, as required. □

B.2 Projection form

Restate Proposition 5.8. *Consider the BRN model $\dot{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{v}(\boldsymbol{x})$ with initial conditions $\boldsymbol{x}(0) = \boldsymbol{x}_0$, and with reactions taking place on L different timescales, admitting the layered decomposition* (5.13)*. Define also the non-stiff dynamics*

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{P}^L \boldsymbol{P}^{L-1} \dots \boldsymbol{P}^2 \boldsymbol{S}^1 \boldsymbol{v}(\tilde{\boldsymbol{x}}), \tag{B.9}$$

where the projections $\boldsymbol{P}^l$ are defined in (5.16)*. The initial conditions of* (5.17) *are given by $\boldsymbol{x}_0 + \sum_{l=2}^L \hat{\boldsymbol{x}}^l(\boldsymbol{x}_0)$, where each $\boldsymbol{\phi}^l(\boldsymbol{x}_0) = \lim_{t\to\infty} \hat{\boldsymbol{x}}^l(t)$ is the steady state of layer l, isolated from the slower layers $k < l$. For $\bar{\boldsymbol{v}}^l$ defined in* (5.15)*, the isolated layers that generate the initial conditions have dynamics*

$$\dot{\hat{\boldsymbol{x}}}^l = \boldsymbol{S}^l \bar{\boldsymbol{v}}^l(\boldsymbol{x}_0 + \hat{\boldsymbol{x}}^l),$$

initialised from $\hat{\boldsymbol{x}}^l(0) = \boldsymbol{0}$. As $\mu \to 0$, the solution $\tilde{\boldsymbol{x}}$ of the dynamics (5.17) *arbitrarily closely approximates the solution $\boldsymbol{x}$ of the original timescale-separated system.*

Proof. Consider the approximated dynamics (B.1) . For $l \geq 2$, each layer's state $\tilde{\boldsymbol{x}}^l = \boldsymbol{\phi}^l(\boldsymbol{x}_0 + \tilde{\boldsymbol{x}}^1 + \dots + \tilde{\boldsymbol{x}}^{l-1})$ has time derivative

$$\dot{\tilde{\boldsymbol{x}}}^l = (\boldsymbol{\phi}^l)' \left[\dot{\tilde{\boldsymbol{x}}}^1 + \dot{\tilde{\boldsymbol{x}}}^2 + \dots + \dot{\tilde{\boldsymbol{x}}}^{l-1}\right].$$

This recursive relationship means that it can be proven by induction that

$$\dot{\tilde{\boldsymbol{x}}}^l = (\boldsymbol{\phi}^l)' \left[\boldsymbol{I} + (\boldsymbol{\phi}^{l-1})'\right] \left[\boldsymbol{I} + (\boldsymbol{\phi}^{l-2})'\right] \cdots \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] \dot{\tilde{\boldsymbol{x}}}^1. \tag{B.10}$$

Indeed, suppose that (B.10) holds for all l with $2 \leq l \leq k$. Then

$$\begin{aligned}
\dot{\tilde{\boldsymbol{x}}}^{k+1} &= (\boldsymbol{\phi}^{k+1})' \left[\dot{\tilde{\boldsymbol{x}}}^1 + \dot{\tilde{\boldsymbol{x}}}^2 + \cdots + \dot{\tilde{\boldsymbol{x}}}^k\right] \\
&= (\boldsymbol{\phi}^{k+1})' \left[\dot{\tilde{\boldsymbol{x}}}^1 + (\boldsymbol{\phi}^2)'\dot{\tilde{\boldsymbol{x}}}^1 + (\boldsymbol{\phi}^3)' \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] \dot{\tilde{\boldsymbol{x}}}^1 + \dots\right] \\
&= (\boldsymbol{\phi}^{k+1})' \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)' + (\boldsymbol{\phi}^3)' \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] + \dots\right] \dot{\tilde{\boldsymbol{x}}}^1 \\
&= (\boldsymbol{\phi}^{k+1})' \left[\boldsymbol{I} + (\boldsymbol{\phi}^3)' + \dots\right] \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] \dot{\tilde{\boldsymbol{x}}}^1 \\
&= \dots \\
&= (\boldsymbol{\phi}^{k+1})' \left[\boldsymbol{I} + (\boldsymbol{\phi}^k)'\right] \left[\boldsymbol{I} + (\boldsymbol{\phi}^{k-1})'\right] \cdots \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] \dot{\tilde{\boldsymbol{x}}}^1,
\end{aligned}$$

as required. The same technique used in this inductive step is used to show that the sum $\tilde{\boldsymbol{x}} = \boldsymbol{x}_0 + \sum_{l=1}^L \tilde{\boldsymbol{x}}^l$ has time derivative

$$\dot{\tilde{\boldsymbol{x}}} = \sum_{l=1}^{L} \dot{\tilde{\boldsymbol{x}}}^l = \left[\boldsymbol{I} + (\boldsymbol{\phi}^L)'\right] \left[\boldsymbol{I} + (\boldsymbol{\phi}^{L-1})'\right] \cdots \left[\boldsymbol{I} + (\boldsymbol{\phi}^2)'\right] \dot{\tilde{\boldsymbol{x}}}^1.$$

It remains to show that each $(\boldsymbol{\phi}^l)'$ is equal to the matrix $\boldsymbol{M}^{l,L}$ given in Definition 5.7.

Recall that $\boldsymbol{\phi}^l(\boldsymbol{z})$ was defined such that $\boldsymbol{0} = \boldsymbol{S}^l \bar{\boldsymbol{v}}^l\big(\boldsymbol{z} + \boldsymbol{\phi}^l(\boldsymbol{z})\big)$. Similarly to Corollary 5.4, the IFT implies that

$$(\boldsymbol{\phi}^l)' = -\boldsymbol{U}^l \left[\boldsymbol{C}^l (\bar{\boldsymbol{v}}^l)' \boldsymbol{U}^l\right]^{-1} \boldsymbol{C}^l (\bar{\boldsymbol{v}}^l)'.$$

For $l = L$, this expression matches the expression for $M^{L,L}$ given in Definition 5.7. For $l < L$, the derivative of $\bar{\boldsymbol{v}}^l(\boldsymbol{z})$ satisfies

$$\begin{aligned}
(\bar{\boldsymbol{v}}^l)' &= \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{z}} \bar{\boldsymbol{v}}^{l+1}\big(\boldsymbol{z} + \boldsymbol{\phi}^{l+1}(\boldsymbol{z})\big) \\
&= (\bar{\boldsymbol{v}}^{l+1})' \left[\boldsymbol{I} + (\boldsymbol{\phi}^{l+1})'\right] = \dots \\
&= \bar{\boldsymbol{v}}' \left[\boldsymbol{I} + (\boldsymbol{\phi}^L)'\right] \cdots \left[\boldsymbol{I} + (\boldsymbol{\phi}^{l+1})'\right].
\end{aligned}$$

By substituting this expression for $(\bar{\boldsymbol{v}}^l)'$ into the formula for $(\boldsymbol{\phi}^l)'$ above, we find that $(\boldsymbol{\phi}^l)' = \boldsymbol{M}^{l,L}$, once more by induction. □

Appendix C

Derivation of Layered Dynamics

Example C.1 Consider the dynamics of the reactions in (7.12) in the form (1.2), given as

$$\begin{bmatrix} \dot{e} \\ \dot{x} \\ \dot{c} \\ \dot{y} \\ \dot{k} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \beta_1 ex \\ \beta_{-1} c \\ \beta_2 c \\ \beta_3 ky \end{bmatrix}$$

from initial conditions $\boldsymbol{x}_0 = (P, x_0, 0, y_0, k_0, 0)^T$. It is easy to show that this system has steady state $\bar{\boldsymbol{x}} = (P, 0, 0, x_0 + y_0 - R, k_0 - R, R)^T$ for $R = \min(x_0 + y_0, k_0)$.

Consider the layering of this system into two layers of dimension 2 and dimension 1 respectively, defined by

$$\boldsymbol{S} = \boldsymbol{S}^1 + \boldsymbol{S}^2 = \begin{bmatrix} -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The ranks of each $\boldsymbol{S}^l$ mean that the dynamics of each layer can be realised in 2 and 1 state variables respectively. This can be achieved, for example, by the $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$ decompositions

$$\boldsymbol{S}^1 = \begin{bmatrix} -1 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{S}^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

The resulting dynamics of each layer are, for states $\boldsymbol{x}'_1 = (x'_{1,1}, x'_{1,2})^T$ and x'_2 respectively, given by

$$\begin{aligned}
\dot{\boldsymbol{x}}'_1 &= \boldsymbol{C}^1 \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^1 \boldsymbol{x}'_1 + \boldsymbol{U}^2 x'_2) \\
&= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 (P - x'_{1,1})(x_0 - x'_{1,1} - x'_{1,2}) - \beta_{-1} x'_{1,1} \\ \beta_2 x'_{1,1} \end{bmatrix}, \\
\dot{x}'_2 &= \boldsymbol{C}^2 \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^1 \boldsymbol{x}'_1 + \boldsymbol{U}^2 x'_2) \\
&= \beta_3 (k_0 - x'_2)(y_0 + x'_{1,2} - x'_2),
\end{aligned}$$

with initial conditions $\boldsymbol{x}'_1(0) = \mathbf{0}$ and $x'_2(0) = 0$. It can be shown that these systems have steady state $\bar{\boldsymbol{x}}_1 = (0, x_0)^T$ and $\bar{x}_2 = R$. By translating these system states to define $\boldsymbol{x}_1 = \boldsymbol{x}'_1 - \bar{\boldsymbol{x}}_1$ and $x_2 = x'_2 - \bar{x}_2$ respectively, the cascade dynamics (7.13) are recovered, with zero steady state and the maps back to the original system state of

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{U}^1(\boldsymbol{x}_1 + \bar{\boldsymbol{x}}_1) + \boldsymbol{U}^2(x_2 + \bar{x}_2),$$

for the originally-modelled state $\boldsymbol{x} = (e, x, \ldots, z)^T$. $\Diamond$

Example C.2 Section 4.5.1 demonstrated the layered decomposition of the large BRN (3.26) with dynamics (3.27). The corresponding stoichiometric matrices are given in (4.18). It has already been shown that this system is a cascade, through the layered dynamics (4.19). Each $\boldsymbol{S}^l$ has rank 3, and therefore each layer can be written in terms of 3-dimensional state vectors in the form (7.17).

The decompositions $\boldsymbol{S}^l = \boldsymbol{U}^l \boldsymbol{C}^l$ of each stoichiometric matrix can be written

$$\boldsymbol{S}^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \\ & \mathbf{0}_{5\times 3} & \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & \\ 0 & -1 & 0 & \mathbf{0}_{3\times 4} \\ 0 & 0 & 1 & \end{bmatrix},$$

$$\boldsymbol{S}^2 = \begin{bmatrix} & \mathbf{0}_{4\times 3} & \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & 1 & -1 & -1 & 0 \\ \mathbf{0}_{3\times 3} & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The resulting dynamics of each layer are, for states $\boldsymbol{x}'_1 = (x'_{1,1}, x'_{1,2}, x'_{1,3})^T$ and $\boldsymbol{x}'_2 = (x'_{2,1}, x'_{2,2}, x'_{2,3})^T$

respectively, given by

$$\begin{aligned}
\dot{\boldsymbol{x}}_1' &= \boldsymbol{C}^1 \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^1\boldsymbol{x}_1' + \boldsymbol{U}^2\boldsymbol{x}_2') \\
&= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1(A_1 + x_{1,1}') \\ \alpha_2(A_2 + x_{1,2}') \\ \alpha_{12}(x_{1,1}' + x_{1,3}')(x_{1,2}' + x_{1,3}') \end{bmatrix}, \\
\dot{\boldsymbol{x}}_2' &= \boldsymbol{C}^2 \boldsymbol{v}(\boldsymbol{x}_0 + \boldsymbol{U}^1\boldsymbol{x}_1' + \boldsymbol{U}^2\boldsymbol{x}_2') \\
&= \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_1(x_0 - x_{1,1}' - x_{1,2}')(x_{1,3}' - x_{2,1}') - \beta_{-1}x_{2,1}' \\ \beta_2 x_{2,1}' \\ \beta_3(k_0 - x_{2,3}')(y_0 + x_{2,2}' - x_{2,3}') \end{bmatrix},
\end{aligned}$$

with initial conditions $\boldsymbol{x}_1'(0) = \boldsymbol{x}_2'(0) = \boldsymbol{0}$. Each of these systems has steady state $\bar{\boldsymbol{x}}_1 = (-A_1, -A_2, P)^T$ and $\bar{\boldsymbol{x}}_2 = (0, x_0, R)$ respectively. By translating these system states to define $\boldsymbol{x}_1 = \boldsymbol{x}_1' - \bar{\boldsymbol{x}}_1$ and $\boldsymbol{x}_2 = \boldsymbol{x}_2' - \bar{\boldsymbol{x}}_2$ respectively, the cascade dynamics (7.17) are recovered, with zero steady state and the maps back to the original system state of

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{U}^1(\boldsymbol{x}_1 + \bar{\boldsymbol{x}}_1) + \boldsymbol{U}^2(\boldsymbol{x}_2 + \bar{\boldsymbol{x}}_2),$$

for the originally-modelled state $\boldsymbol{x} = (p_1, p_2, \ldots, z)^T$. $\Diamond$

References

[1] J. Ackermann, J. Einloft, J. Nöthen, and I. Koch. Reduction techniques for network validation in systems biology. *Journal of Theoretical Biology*, **315**:71–80, 2012.

[2] A. A. Ahmadi and P. A. Parrilo. Towards scalable algorithms with formal guarantees for Lyapunov analysis of control systems via algebraic optimization. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2014.

[3] T. Aittokallio and B. Schwikowski. Graph-based methods for analysing networks in cell biology. *Briefings In Bioinformatics*, **7**(3):243–255, 2006.

[4] R. P. Alexander, P. M. Kim, T. Emonet, and M. B. Gerstein. Understanding modularity in molecular networks requires dynamics. *Science Signalling*, **2**(81):pe44, 2009.

[5] R. J. R. Algar, T. Ellis, and G.-B. Stan. Modelling the burden caused by gene expression: an in silico investigation into the interactions between synthetic gene circuits and their chassis cell. *arXiv preprint arXiv:1309.7798*, 2013.

[6] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, **8**(6):450–461, 2007.

[7] J. C. Anderson, C. A. Voigt, and A. P. Arkin. Environmental signal integration by a modular AND gate. *Molecular Systems Biology*, **3**:133, 2007.

[8] J. Anderson, Y.-C. Chang, and A. Papachristodoulou. Model decomposition and reduction tools for large-scale networks in systems biology. *Automatica*, **47**(6):1165–1174, 2011.

[9] J. Anderson and A. Papachristodoulou. On validation and invalidation of biological models. *BMC Bioinformatics*, **10**(1):132, 2009.

[10] J. Anderson and A. Papachristodoulou. A decomposition technique for nonlinear dynamical system analysis. *IEEE Transactions on Automatic Control*, **57**(6):1516–1521, 2012.

[11] M. Andrec, B. N. Kholodenko, R. M. Levy, and E. D. Sontag. Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. *Journal of Theoretical Biology*, **232**:427–441, 2005.

[12] D. ANGELI. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, **47**(3):410–421, 2002.

[13] D. ANGELI, P. D. LEENHEER, AND E. SONTAG. Graph-theoretic characterizations of monotonicity of chemical networks in reaction coordinates. *Journal of Mathematical Biology*, **61**:581–616, 2010.

[14] D. ANGELI AND E. D. SONTAG. Monotone control systems. *IEEE Transactions on Automatic Control*, **48**(10):1684–1698, 2003.

[15] A. C. ANTOULAS. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[16] M. APRI, M. DE GEE, AND J. MOLENAAR. Complexity reduction preserving dynamical behavior of biochemical networks. *Journal of Theoretical Biology*, **304**:16–26, 2012.

[17] J. A. J. ARPINO, E. J. HANCOCK, J. ANDERSON, M. BARAHONA, G.-B. V. STAN, A. PAPACHRISTODOULOU, AND K. POLIZZI. Tuning the dials of Synthetic Biology. *Microbiology*, **159**:1236–1253, 2013.

[18] L. BAKULE. Decentralized control: An overview. *Annual Reviews in Control*, **32**:87–98, 2008.

[19] C. J. BASHOR, A. A. HOROWITZ, S. G. PEISAJOVICH, AND W. A. LIM. Rewiring cells: Synthetic biology as a tool to interrogate the organizational principles of living systems. *Annual Review of Biophysics*, **39**:515–537, 2010.

[20] A. BECSKEI, B. SÉRAPHIN, AND L. SERRANO. Positive feedback in eukaryotic gene networks: Cell differentiation by graded to binary response conversion. *EMBO Journal*, **20**:2528–2535, 2001.

[21] M. BEHAR, H. G. DOHLMAN, AND T. C. ELSTON. Kinetic insulation as an effective mechanism for achieving pathway specificity in intracellular signaling networks. *PNAS*, **104**(41):16146–16151, 2007.

[22] P. BENDOTTI AND C. L. BECK. On the role of LFT model reduction methods in robust controller synthesis for a pressurized water reactor. *IEEE Transactions on Control Systems Technology*, **7**(2):248–257, 1999.

[23] M. R. BENNETT, D. VOLFSON, L. TSIMRING, AND J. HASTY. Transient dynamics of genetic regulatory networks. *Biophysical Journal*, **92**:3501–3512, 2007.

[24] B. BESSELINK, N. VAN DE WOUW, AND H. NIJMEIJER. Model reduction for nonlinear systems with incremental gain or passivity properties. *Automatica*, **49**:861–872, 2013.

[25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10):P10008, 2008.

[26] N. Blüthgen and H. Herzel. How robust are switches in intracellular signaling cascades? *Journal of Theoretical Biology*, **225**(3):293–300, 2003.

[27] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, **424**:175–308, 2006.

[28] J. A. Borghans, R. J. De Boer, and L. A. Segel. Extending the quasi-steady state approximation by changing variables. *Bulletin Of Mathematical Biology*, **58**(1):43–63, 1996.

[29] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[30] R. Breitling, D. Gilbert, M. Heiner, and R. Orton. A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. *Briefings in Bioinformatics*, **9**(5):404–421, 2008.

[31] K. Brenner, L. You, and F. H. Arnold. Engineering microbial consortia: a new frontier in synthetic biology. *Trends in Biotechnology*, **26**(9):483–489, 2008.

[32] F. J. Bruggeman, J. L. Snoep, and H. V. Westerhoff. Control, responses and modularity of cellular regulatory networks: a control analysis perspective. *IET Systems Biology*, **2**(6):397–410, 2008.

[33] F. J. Bruggeman, N. Blüthgen, and H. V. Westerhoff. Noise management by molecular networks. *PLoS Computational Biology*, **5**(9):e1000506, 2009.

[34] F. J. Bruggeman and H. V. Westerhoff. The nature of systems biology. *Trends in Microbiology*, **15**(1):45–50, 2007.

[35] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, **464**(7291):1025–1028, 2010.

[36] R. Bundschuh, F. Hayot, and C. Jayaprakash. Fluctuations and slow variables in genetic networks. *Biophysical Journal*, **84**:1606–1615, 2003.

[37] G. Buzi, U. Topcu, and J. C. Doyle. Analysis of autocatalytic networks in biology. *Automatica*, **47**:1123–1130, 2011.

[38] G. D. Byrne and A. C. Hindmarsh. Stiff ode solvers: A review of current and coming attractions. *Journal of Computational Physics*, **70**(1):1–62, 1987.

[39] J. M. Callura, C. R. Cantor, and J. J. Collins. Genetic switchboard for synthetic biology applications. *PNAS*, **109**:5850–5855, 2012.

[40] S. Cardinale and A. P. Arkin. Contextualizing context for synthetic biology—identifying causes of failure of synthetic biological systems. *Biotechnology Journal*, **7**:856–866, 2012.

[41] F. A. Chandra, G. Buzi, and J. C. Doyle. Glycolytic oscillations and limits on robust efficiency. *Science*, **333**(6039):187–192, 2011.

[42] J. Chen and B. Yuan. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, **22**(18):2283–2290, 2006.

[43] L. Chen, R. Wang, C. Li, and K. Aihara. *Modeling Biomolecular Networks in Cells*. Springer, 2010.

[44] W. W. Chen, B. Schoeberl, P. J. Jasper, M. Niepel, U. B. Nielsen, D. A. Lauffenburger, and P. K. Sorger. Input–output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. *Molecular Systems Biology*, **5**:239, 2009.

[45] G. Chesi. On the gap between positive polynomials and SOS of polynomials. *IEEE Transactions on Automatic Control*, **52**(6):1066–1072, 2007.

[46] G. Chesi. LMI techniques for optimization over polynomials in control: a survey. *IEEE Transactions on Automatic Control*, **55**(11):2500–2510, 2010.

[47] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, **95**(1):255–312, 2007.

[48] A. Ciliberto, F. Capuani, and J. J. Tyson. Modeling networks of coupled enzymatic reactions using the total quasi-steady state approximation. *PLoS Computational Biology*, **3**(3):e45, 2007.

[49] A. Colman-Lerner, A. Gordon, E. Serra, T. Chin, O. Resnekov, D. Endy, C. G. Pesce, and R. Brent. Regulated cell-to-cell variation in a cell-fate decision system. *Nature*, **437**(7059):699–706, 2005.

[50] N. A. Cookson, W. H. Mather, T. Danino, O. Mondragón-Palomino, R. J. Williams, L. S. Tsimring, and J. Hasty. Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Molecular Systems Biology*, **7**:561, 2011.

[51] A. Cornish-Bowden. *Fundamentals of Enzyme Kinetics*. Portland Press, 3rd edition, 2004.

[52] E. J. Crampin, S. Schnell, and P. E. McSharry. Mathematical and computational techniques to deduce complex biochemical reaction mechanisms. *Progress in Biophysics and Molecular Biology*, **86**:77–112, 2004.

[53] F. DABBENE AND D. HENRION. Minimum volume semialgebraic sets for robust estimation. *arXiv preprint arXiv:1210.3183*, 2012.

[54] T. L. DEANS, C. R. CANTOR, AND J. J. COLLINS. A tunable genetic switch based on RNAi and repressor proteins for regulating gene expression in mammalian cells. *Cell*, **130**:363–372, 2007.

[55] D. DEL VECCHIO. A control theoretic framework for modular analysis and design of biomolecular networks. *IFAC Reviews in Control*, **Accepted**:TBC, 2013.

[56] D. DEL VECCHIO, A. J. NINFA, AND E. D. SONTAG. Modular cell biology: retroactivity and insulation. *Molecular Systems Biology*, **4**:161, 2008.

[57] D. DEL VECCHIO AND E. D. SONTAG. Engineering principles in bio-molecular systems: from retroactivity to modularity. *European Journal of Control*, **15**(3–4):389–397, 2009.

[58] J. E. DUEBER, B. J. YEH, K. CHAK, AND W. A. LIM. Reprogramming control of an allosteric signaling switch through modular recombination. *Science*, **301**:1904–1908, 2003.

[59] G. E. DULLERUD AND F. PAGANINI. *A Course in Robust Control Theory: a Convex Approach*. Springer, 2000.

[60] C. EBENBAUER, T. RAFF, AND F. ALLGÖWER. Dissipation inequalities in systems theory: An introduction and recent results. In *Invited Lectures of the International Congress on Industrial and Applied Mathematics*, **2007**, pages 23–42, 2009.

[61] H. EL-SAMAD, M. KHAMMASH, L. PETZOLD, AND D. GILLESPIE. Stochastic modelling of gene regulatory networks. *International Journal of Robust and Nonlinear Control*, **15**(15):691–711, 2005.

[62] M. B. ELOWITZ AND S. LEIBLER. A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**:335–338, 2000.

[63] D. F. ENNS. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, **23**, pages 127–132. IEEE, 1984.

[64] J. A. FAX AND R. M. MURRAY. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, **49**(9):1465–1476, 2004.

[65] D. A. FELL. Metabolic control analysis: a survey of its theoretical and experimental development. *Biochemical Journal*, **286**:313–330, 1992.

[66] X.-J. FENG AND H. RABITZ. Optimal identification of biochemical reaction networks. *Biophysical Journal*, **86**(3):1270–1281, 2004.

[67] S. FORTUNATO. Community detection in graphs. *Physics Reports*, **486**:75–174, 2010.

[68] E. FUNG, W. W. WONG, J. K. SUEN, T. BULTER, S. GU LEE, AND J. C. LIAO. A synthetic gene–metabolic oscillator. *Nature*, **435**:118–122, 2005.

[69] J. GAO, S. V. BULDYREV, H. E. STANLEY, AND S. HAVLIN. Networks formed from interdependent networks. *Nature Physics*, **8**:40–48, 2012.

[70] T. S. GARDNER, C. R. CANTOR, AND J. J. COLLINS. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, **403**:339–342, 2000.

[71] P. GENNEMARK, B. NORDLANDER, S. HOHMANN, AND D. WEDELIN. A simple mathematical model of adaptation to high osmolarity in yeast. *In Silico Biology*, **6**:193–214, 2006.

[72] Z. P. GERDTZEN, P. DAOUTIDIS, AND W.-S. HU. Non-linear reduction for kinetic models of metabolic reaction networks. *Metabolic Engineering*, **6**:140–154, 2004.

[73] D. T. GILLESPIE. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, **81**:2340–2361, 1977.

[74] K. GLOVER. All optimal Hankel-norm approximations of linear multivariable systems and their L^{∞}-error bounds. *International Journal of Control*, **39**(6):1115–1193, 1984.

[75] K.-I. GOH, B. KAHNG, AND K.-H. CHO. Sustained oscillations in extended genetic oscillatory systems. *Biophysical Journal*, **67**:4270–4276, 2008.

[76] A. GOLDBETER AND D. E. KOSHLAND. An amplified sensitivity arising from covalent modification in biological systems. *PNAS*, **78**(11):6840–6844, 1981.

[77] E. GONÇALVES, J. BUCHER, A. RYLL, J. NIKLAS, K. MAUCH, S. KLAMT, M. ROCHA, AND J. SAEZ-RODRIGUEZ. Bridging the layers: towards integration of signal transduction, regulation and metabolism into mathematical models. *Molecular Biosystems*, **9**(7):1576–1583, 2013.

[78] J. GOUTSIAS. Classical versus stochastic kinetics modeling of biochemical reaction systems. *Biophysical Journal*, **92**(7):2350–2365, 2007.

[79] J. GUNAWARDENA. Chemical reaction network theory for in-silico biologists. Notes available for download at `http://vcp.med.harvard.edu/papers/crnt.pdf`, 2003.

[80] A. GUPTA, C. BRIAT, AND M. KHAMMASH. A scalable computational framework for establishing long-term behavior of stochastic reaction networks. *PLoS Computational Biology*, **10**(6):e1003669, 2014.

[81] R. N. GUTENKUNST, J. J. WATERFALL, F. P. CASEY, K. S. BROWN, C. R. MYERS, AND J. P. SETHNA. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, **3**(10):e189, 2007.

[82] A. GYORGY AND D. DEL VECCHIO. Limitations and trade-offs in gene expression due to competition for shared cellular resources. In *Proceedings of the IEEE Conference on Decision and Control*, 2014.

[83] A. GYORGY AND D. DEL VECCHIO. Modular composition of gene transcription networks. *PLoS Computational Biology*, **10**:e1003486, 2014.

[84] W. M. HADDAD AND V. CHELLABOINA. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton University Press, 2008.

[85] A. HAMADEH, E. SONTAG, AND B. INGALLS. Response time re-scaling and Weber's law in adapting biological systems. In *Proceedings of the American Control Conference (ACC)*, 2013.

[86] J. HAMON, P. JENNINGS, AND F. Y. BOIS. Systems biology modeling of omics data: effect of cyclosporine a on the nrf2 pathway in human renal cells. *BMC Systems Biology*, **8**(1):76, 2014.

[87] E. J. HANCOCK AND A. PAPACHRISTODOULOU. Structured sum of squares for networked systems analysis. In *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 7236–7241. IEEE, 2011.

[88] R. HANNEMANN-TAMÁS, A. GÁBOR, G. SZEDERKÉNYI, AND K. M. HANGOS. Model complexity reduction of chemical reaction networks using mixed-integer quadratic programming. *Computers and Mathematics with Applications*, 2012.

[89] F. HE, V. FROMION, AND H. V. WESTERHOFF. (Im)Perfect robustness and adaptation of metabolic networks subject to metabolic and gene-expression regulation: Marrying control engineering with metabolic control analysis. *BMC Systems Biology*, **7**:131, 2013.

[90] J.-H. HOFMEYR AND H. V. WESTERHOFF. Building the cellular puzzle: control in multilevel reaction networks. *Journal of Theoretical Biology*, **208**:261–285, 2001.

[91] F. HORN AND R. JACKSON. General mass action kinetics. *Archive for Rational Mechanics and Analysis*, **47**(2):81–116, 1972.

[92] R. A. HORN AND C. R. JOHNSON. *Matrix Analysis.* Cambridge University Press, 1985.

[93] M. HUCKA, A. FINNEY, H. M. SAURO, H. BOLOURI, J. C. DOYLE, H. KITANO, A. P. ARKIN, B. J. BORNSTEIN, D. BRAY, A. CORNISH-BOWDEN, ET AL. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4):524–531, 2003.

[94] F. HYNNE, S. DANØ, AND P. SØRENSEN. Full-scale model of glycolysis in *Saccharomyces cerevisiae. Biophysical Chemistry*, **94**(1–2):121–163, 2001.

[95] M. Ikeda and D. D. Šiljak. Lotka-Volterra equations: Decomposition, stability, and structure. *Journal of Mathematical Biology*, **9**:65–83, 1980.

[96] M. Ikeda, D. D. Šiljak, and D. E. White. An inclusion principle for dynamic systems. *IEEE Transactions on Automatic Control*, **29**(3):244–249, 1984.

[97] B. P. Ingalls and P. A. Iglesias, editors. *Control Theory and Systems Biology*. MIT Press, 2010.

[98] T. Ishizaki, H. Sandberg, K. H. Johansson, K. Kashima, J. ichi Imura, and K. Aihara. Structured model reduction of interconnected linear systems based on singular perturbation. In *Proceedings of the American Control Conference (ACC)*, 2013.

[99] N. Jamshidi and B. O. Palsson. Top-down analysis of temporal hierarchy in biochemical reaction networks. *PLoS Computational Biology*, **4**(9):e1000177, 2008.

[100] L. G. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney. Think locally, act locally: The detection of small, medium-sized, and large communities in large networks. arXiv preprint arXiv:1403.3795, 2014.

[101] R. L. Karp, M. P. Millán, T. Dasgupta, A. Dickenstein, and J. Gunawardena. Complex-linear invariants of biochemical networks. *Journal of Theoretical Biology*, **311**:130–138, 2012.

[102] K. J. Kauffman, P. Prakash, and J. S. Edwards. Advances in flux balance analysis. *Current Opinion in Biotechnology*, **14**:491–496, 2003.

[103] A. S. Khalil and J. J. Collins. Synthetic biology: Applications come of age. *Nature Reviews Genetics*, **11**:367–379, 2010.

[104] B. N. Kholodenko, A. Kiyatkin, F. J. Bruggeman, E. Sontag, and H. V. Westerhoff. Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *PNAS*, **99**(20):12841–12846, 2002.

[105] B. N. Kholodenko and E. D. Sontag. Determination of functional network structure from local parameter dependence data. arXiv:physics/0205003 [physics.bio-ph], 2002.

[106] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. Porter. Multilayer networks. *Journal of Complex Networks*, 2014.

[107] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Computational Biology*, **5**(5):e1000385, 2009.

[108] W. Klonowski. Simplifying principles for chemical and enzyme reaction kinetics. *Biophysical Chemistry*, **18**(2):73–87, 1983.

[109] S. Klumpp, Z. Zhang, and T. Hwa. Growth rate-dependent global effects on gene expression in bacteria. *Cell*, **139**:1366–1375, 2009.

[110] P. V. Kokotovic, J. J. Allemong, J. R. Winkelman, and J. H. Chow. Singular perturbation and iterative separation of time scales. *Automatica*, **16**:23–33, 1980.

[111] N. Kottenstette, M. J. McCourt, M. Xia, V. Gupta, and P. J. Antsaklis. On relationships among passivity, positive realness, and dissipativity in linear systems. *Automatica*, **50**(4):1003–1016, 2014.

[112] B. P. Kramer, A. U. Viretta, M. Daoud-El Baba, D. Aubel, W. Weber, and M. Fussenegger. An engineered epigenetic transgene switch in mammalian cells. *Nature Biotechnology*, **22**:867–870, 2004.

[113] R. Krüger and R. Heinrich. Model reduction and analysis of robustness for the Wnt/β-Catenin signal transduction pathway. *Genome Informatics*, **15**(1):138–148, 2004.

[114] H. Kurata, H. El-Samad, R. Iwasaki, H. Ohtake, J. C. Doyle, I. Grigorova, C. A. Gross, and M. Khammash. Module-based analysis of robustness tradeoffs in the heat shock response system. *PLoS Computational Biology*, **2**(7):e59, 2006.

[115] M. Lang, S. Summers, and J. Stelling. Cutting the wires: Modularization of cellular networks for experimental design. *Biophysical Journal*, **106**:321–331, 2014.

[116] C. H. Lee and H. G. Othmer. A multi-time-scale analysis of chemical reaction networks: Deterministic systems. *Journal of Mathematical Biology*, **60**(3):387–450, 2010.

[117] C. Li, M. Donizelli, N. Rodriguez, H. Dharuri, L. Endler, V. Chelliah, L. Li, E. He, A. Henry, M. I. Stefan, J. L. Snoep, M. Hucka, N. Le Novère, and C. Laibe. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology*, **4**:92, Jun 2010.

[118] Y. Li, G. Dwivedi, W. Huang, M. L. Kemp, and Y. Yi. Quantification of degeneracy in biological systems for characterization of functional interactions between modules. *Journal of Theoretical Biology*, **302**(1):29–38, 2012.

[119] C.-C. Lin, Y.-J. Chen, C.-Y. Chen, Y.-J. Oyang, H.-F. Juan, and H.-C. Huang. Crosstalk between transcription factors and microRNAs in human protein interaction network. *BMC Systems Biology*, **6**:18, 2012.

[120] Z.-P. Liu, Y. Wang, X.-S. Zhang, and L. Chen. Network-based analysis of complex diseases. *IET Systems Biology*, **6**(1):22–33, 2012.

[121] C. Lobry and T. Sari. Singular perturbation methods in control theory. *Contrôle non linéaire et Applications, Travaux en Cours*, **64**:151–177, 2005.

[122] T. K. LU, A. S. KHALIL, AND J. J. COLLINS. Next-generation synthetic gene networks. *Nature Biotechnology*, **27**(12):1139–1150, 2009.

[123] S. MANDAL AND R. SARPESHKAR. Log-domain circuit models of chemical reactions. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2697–2700. IEEE, 2009.

[124] G. MARSAGLIA AND G. P. H. STYAN. Equalities and inequalities for ranks of matrices. *Linear and Multilinear Algebra*, **2**:269–292, 1974.

[125] R. P. MASON AND A. PAPACHRISTODOULOU. Chordal sparsity, decomposing sdps and the lyapunov equation. In *Proceedings of the American Control Conference (ACC)*, 2014.

[126] M. N. MCCLEAN, A. MODY, J. R. BROACH, AND S. RAMANATHAN. Cross-talk and decision making in MAP kinase pathways. *Nature Genetics*, **39**(3):409–414, 2007.

[127] A. MEGRETSKI AND A. RANTZER. System analysis via Integral Quadratic Constraints. *IEEE Transactions on Automatic Control*, **42**(6):819–830, 1997.

[128] C. MEISSEN, L. LESSARD, M. ARCAK, AND A. PACKARD. Performance certification of interconnected nonlinear systems using ADMM. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2014.

[129] C. MEISSEN, L. LESSARD, AND A. PACKARD. Performance certification of interconnected systems using decomposition techniques. In *Proceedings of the American Control Conference*, 2014.

[130] B. MÉLYKÚTI, K. BURRAGE, AND K. C. ZYGALAKIS. Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical Langevin equation. *Journal of Chemical Physics*, **132**:164109, 2010.

[131] R. MILO, S. SHEN-ORR, S. ITZKOVITZ, N. KASHTAN, D. CHKLOVSKII, AND U. ALON. Network motifs: Simple building blocks of complex networks. *Science*, **298**(5594):824–827, 2002.

[132] P. J. MUCHA, T. RICHARDSON, K. MACON, M. A. PORTER, AND J.-P. ONNELA. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, **328**:876–878, 2010.

[133] J. D. MURRAY. *Mathematical Biology: an Introduction.* Springer, 2002.

[134] T. J. NEWMAN. Life and death in biophysics. *Physical Biology*, **8**(1):010201, 2011.

[135] K. OISHI AND E. KLAVINS. Biomolecular implementation of linear I/O systems. *IET Systems Biology*, **5**(4):252–260, 2011.

[136] D. A. OYARZÚN, M. CHAVES, AND M. HOFF-HOFFMEYER-ZLOTNIK. Multistability and oscillations in genetic control of metabolism. *Journal of Theoretical Biology*, **295**:139–153, 2012.

[137] B. O. PALSSON. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[138] B. O. PALSSON. *Systems Biology: Simulation of Dynamic Network States*. Cambridge University Press, 2011.

[139] A. PAPACHRISTODOULOU, J. ANDERSON, G. VALMORBIDA, S. PRAJNA, P. SEILER, AND P. A. PARRILO. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB:*. `http://arxiv.org/abs/1310.4716`, 2013. Available from `http://www.eng.ox.ac.uk/control/sostools`.

[140] 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 (CDC)*, 2002.

[141] A. PAPACHRISTODOULOU AND S. PRAJNA. A Tutorial on Sum of Squares techniques for systems analysis. In *Proceedings of the American Control Conference (ACC)*, 2005.

[142] J. A. PAPIN, J. L. REED, AND B. O. PALSSON. Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. *Trends in Biochemical Sciences*, **29**(12):641–647, 2004.

[143] P. A. PARRILO. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, Caltech, Pasadena, CA, 2000.

[144] J. M. PEDRAZA AND A. VAN OUDENAARDEN. Noise propagation in gene networks. *Science*, **307**(5717):1965–1969, 2005.

[145] M. A. PORTER, J.-P. ONNELA, AND P. J. MUCHA. Communities in networks. *Notices of the AMS*, **56**(9):1082–1097, 2009.

[146] S. PRAJNA AND H. SANDBERG. On model reduction of polynomial dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2005.

[147] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Synthetic biology: A control engineering perspective. In *Proceedings of the European Control Conference (ECC)*, 2014.

[148] T. P. PRESCOTT AND A. GYORGY. Bounding the effect of retroactivity in the presence of parameter uncertainty. In *Proceedings of American Control Conference (ACC)*, 2015. Paper accepted.

[149] T. P. PRESCOTT, M. LANG, AND A. PAPACHRISTODOULOU. Quantification of interactions between dynamic cellular network functionalities by cascaded layering. *PLoS Computational Biology*, **11**(5):e1004235, 2015.

[150] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Guaranteed error bounds for structured complexity reduction of biochemical networks. *Journal of Theoretical Biology*, **302**:172–182, 2012.

[151] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Layering in networks: The case of biochemical systems. In *Proceedings of the American Control Conference (ACC)*, pages 4544–4549, 2013.

[152] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Layered decomposition for the model order reduction of timescale separated biochemical reaction networks. *Journal of Theoretical Biology*, **356**:113–122, 2014.

[153] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Signal propagation across layered biochemical networks. In *Proceedings of the American Control Conference (ACC)*, pages 3399–3404, 2014.

[154] T. P. PRESCOTT AND A. PAPACHRISTODOULOU. Structured storage functions for cascaded systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2014.

[155] P. E. PURNICK AND R. WEISS. The second wave of synthetic biology: From modules to systems. *Nature Reviews Molecular Cell Biology*, **10**:410–422, 2009.

[156] O. RADULESCU, A. N. GORBAN, A. ZINOVYEV, AND V. NOEL. Reduction of dynamical biochemical reactions networks in computational biology. *Frontiers in Genetics*, **3**:131, 2012.

[157] J. O. RAMSAY, G. HOOKER, D. CAMPBELL, AND J. CAO. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **69**(5):741–796, 2007.

[158] U. RAND, M. RINAS, J. SCHWERK, G. NÖHREN, M. LINNES, A. KRÖGER, M. FLOSSDORF, K. KÁLY-KULLAI, H. HAUSER, T. HÖFER, AND M. KÖSTER. Multi-layered stochasticity and paracrine signal propagation shape the type-I interferon response. *Molecular Systems Biology*, **8**:584, 2012.

[159] S. RAO, A. VAN DER SCHAFT, AND B. JAYAWARDHANA. A graph-theoretical approach for the analysis and model reduction of complex-balanced chemical reaction networks. *Journal of Mathematical Chemistry*, **51**(9):2401–2422, 2013.

[160] A. RAUE, J. KARLSSON, M. P. SACCOMANI, M. JIRSTRAND, AND J. TIMMER. Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, **30**(10):1440–1448, 2014.

[161] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, **52**(3):471–501, 2010.

[162] B. Recht, W. Xu, and B. Hassibi. Null space conditions and thresholds for rank minimization. *Mathematical Programming*, **127**(1):175–202, 2011.

[163] B. G. Romanchuk and M. R. James. Characterization of the L_p incremental gain for nonlinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, 1996.

[164] S. Rosenfeld. Patterns of stochastic behavior in dynamically unstable high-dimensional biochemical networks. *Gene Regulation & Systems Biology*, **3**:1–10, 2009.

[165] J. Saez-Rodriguez, S. Gayer, M. Ginkel, and E. D. Gilles. Automatic decomposition of kinetic models of signaling networks minimizing the retroactivity among modules. *Bioinformatics*, **24**(16):i213–i219, 2008.

[166] J. Saez-Rodriguez, A. Kremling, H. Conzelmann, K. Bettenbrock, and E. D. Gilles. Modular analysis of signal transduction networks. *IEEE Control Systems Magazine*, **24**(4):35–52, 2004.

[167] J. Saez-Rodriguez, A. Kremling, and E. D. Gilles. Dissecting the puzzle of life: modularization of signal transduction networks. *Computers & Chemical Engineering*, **29**(3):619–629, 2005.

[168] A. Samal and S. Jain. The regulatory network of *E. coli* metabolism as a Boolean dynamical system exhibits both homeostasis and flexibility of response. *BMC Systems Biology*, **2**:21, 2008.

[169] H. Sandberg and R. M. Murray. Model reduction of interconnected linear systems. *Optimal Control Applications and Methods*, **30**:225–245, 2009.

[170] H. Sandberg. An extension to balanced truncation with application to structured model reduction. *IEEE Transactions on Automatic Control*, **55**(4):1038–1043, 2010.

[171] J. Schaber, B. Kofahl, A. Kowald, and E. Klipp. A modelling approach to quantify dynamic crosstalk between the pheromone and the starvation pathway in baker's yeast. *FEBS Journal*, **273**(15):3520–3533, 2006.

[172] J. M. A. Scherpen. Balancing for nonlinear systems. *Systems & Control Letters*, **21**:143–153, 1993.

[173] D. Schittler, T. Jouini, F. Allgöwer, and S. Waldherr. Generalization of the construction method for multistability-equivalent gene regulatory networks to systems with multi-input multi-output loopbreaking. arXiv:1312.7250 [math.DS], 2014.

[174] H. SCHMIDT AND M. JIRSTRAND. Systems biology toolbox for matlab: a computational platform for research in systems biology. *Bioinformatics*, **22**(4):514–515, 2006.

[175] S. SCHNELL AND P. K. MAINI. A Century of Enzyme Kinetics: Reliability of the K_M and V_{max} Estimates. *Comments on Theoretical Biology*, **8**:169–187, 2003.

[176] S. SCHUSTER, T. DANDEKAR, AND D. A. FELL. Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology*, **17**:53–60, 1999.

[177] S. SCHUSTER AND C. HILGETAG. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, **2**(2):165–182, 1994.

[178] S. SCHUSTER, D. KAHN, AND H. V. WESTERHOFF. Modular analysis of the control of complex metabolic pathways. *Biophysical Chemistry*, **48**(1):1–17, 1993.

[179] B. SCHWIKOWSKI, P. UETZ, AND S. FIELDS. A network of protein–protein interactions in yeast. *Nature Biotechnology*, **18**:1257–1261, 2000.

[180] L. A. SEGEL AND M. SLEMROD. The quasi-steady-state assumption: a case study in perturbation. *SIAM Review*, **31**(3):446–477, 1989.

[181] P. SEILER. Nonlinear stability analysis with dissipation inequalities and Integral Quadratic Constraints. *IEEE Transactions on Automatic Control*, 2013. Submitted.

[182] C. SHI, C.-H. HUANG, P. N. DEVREOTES, AND P. A. IGLESIAS. Interaction of motility, directional sensing, and polarity modules recreates the behaviors of chemotaxing cells. *PLoS Computational Biology*, **9**(7):e1003122, 2013.

[183] T. SHIFRIN. *Multivariable Mathematics: Linear Algebra, Multivariable Calculus, and Manifolds.* John Wiley and Sons, Hoboken, New Jersey, 2005.

[184] W. SHOU, S. RAM, AND J. M. VILAR. Synthetic cooperation in engineered yeast populations. *PNAS*, **104**(6):1877–1882, 2007.

[185] D. SIEGAL-GASKINS, V. NOIREAUX, AND R. M. MURRAY. Biomolecular resource utilization in elementary cell-free gene circuits. In *Proceedings of the American Control Conference (ACC)*, pages 1531–1536. IEEE, 2013.

[186] H. SIVAKUMAR, S. R. PROULX, AND J. P. HESPANHA. Modular decomposition and analysis of biological networks. *arXiv preprint arXiv:1412.0742*, 2014.

[187] J. M. SKERKER, B. S. PERCHUK, A. SIRYAPORN, E. A. LUBIN, O. ASHENBERG, M. GOULIAN, AND M. T. LAUB. Rewiring the specificity of two-component signal transduction systems. *Cell*, **133**:1043–1054, 2008.

[188] A. L. SLUSARCZYK, A. LIN, AND R. WEISS. Foundations for the design and implementation of synthetic genetic circuits. *Nature Reviews Genetics*, **13**:406–420, 2012.

[189] E. D. SONTAG. A technique for determining the signs of sensitivities of steady states in chemical reaction networks. *IET Systems Biology*, **8**(6):251–267, 2014.

[190] E. D. SONTAG, A. KIYATKIN, AND B. N. KHOLODENKO. Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics*, **20**(12):1877–1886, 2004.

[191] R. SRIVASTAVA, L. YOU, J. SUMMERS, AND J. YIN. Stochastic vs. deterministic modeling of intracellular viral kinetics. *Journal of Theoretical Biology*, **218**(3):309–321, 2002.

[192] N. J. STANFORD, T. LUBITZ, K. SMALLBONE, E. KLIPP, P. MENDES, AND W. LIEBERMEISTER. Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, **8**(11):e79195, 2013.

[193] L. D. STEIN. Integrating biological databases. *Nature Reviews Genetics*, **4**(5):337–345, 2003.

[194] J. STELLING. Mathematical models in microbial systems biology. *Current Opinion in Microbiology*, **7**:513–518, 2004.

[195] J. STELLING, S. KLAMT, K. BETTENBROCK, S. SCHUSTER, AND E. D. GILLES. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, **420**:190–193, 2002.

[196] J. STRICKER, S. COOKSON, M. R. BENNETT, W. H. MATHER, L. S. TSIMRING, AND J. HASTY. A fast, robust and tunable synthetic gene oscillator. *Nature*, **456**:516–519, 2008.

[197] J. STURM. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **11–12**:625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`.

[198] E. SUMMERS AND A. PACKARD. L_2 gain verification for interconnections of locally stable systems using Integral Quadratic Constraints. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2010.

[199] Z. Z. SUN, C. A. HAYES, J. SHIN, F. CASCHERA, R. M. MURRAY, AND V. NOIREAUX. Protocols for implementing an *Escherichia coli* based TX-TL cell-free expression system for synthetic biology. *Journal of Visualized Experiments: JoVE*, **79**:e50762, 2013.

[200] M. SUNNÅKER, G. CEDERSUND, AND M. JIRSTRAND. A method for zooming of nonlinear models of biochemical systems. *BMC Systems Biology*, **5**:140, 2011.

[201] I. SUROVTSOVA, N. SIMUS, K. HUBNER, S. SAHLE, AND U. KUMMER. Simplification of biochemical models: A general approach based on the analysis of the impact of individual species and reactions on the systems dynamics. *BMC Systems Biology*, **6**:14, 2012.

[202] M. TERZER. *Large scale methods to enumerate extreme rays and elementary modes.* PhD thesis, ETH Zürich, 2009. Software from `http://www.csb.ethz.ch/tools/efmtool`.

[203] A. H. Y. TONG, G. LESAGE, G. D. BADER, H. DING, H. XU, X. XIN, J. YOUNG, G. F. BERRIZ, R. L. BROST, M. CHANG, ET AL. Global mapping of the yeast genetic interaction network. *Science*, **303**(5659):808–813, 2004.

[204] D. TSUBAKINO AND S. HARA. Eigenvector-based intergroup connection of low rank for hierarchical multi-agent dynamical systems. *Systems & Control Letters*, **61**(2):354–361, 2012.

[205] A. R. TZAFRIRI. Michaelis–Menten kinetics at high enzyme concentrations. *Bulletin of Mathematical Biology*, **65**:1111–1129, 2003.

[206] R. R. VALLABHAJOSYULA, V. CHICKARMANE, AND H. M. SAURO. Conservation analysis of large biochemical networks. *Bioinformatics*, **22**(3):346–353, 2006.

[207] A. VANDENDORPE AND P. VAN DOOREN. Model reduction of interconnected systems. In *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, 2008.

[208] A. F. VILLAVERDE AND J. R. BANGA. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of The Royal Society Interface*, **11**(91):20130505, 2014.

[209] K. E. WELLEN AND C. B. THOMPSON. A two-way street: reciprocal regulation of metabolism and signalling. *Molecular Cell Biology*, **13**:270–276, 2012.

[210] D. J. WILKINSON. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, **10**(2):122–133, 2009.

[211] D. M. WILKINSON AND B. A. HUBERMAN. A method for finding communities of related genes. *PNAS*, **101**(suppl 1):5241–5248, 2004.

[212] J. C. WILLEMS. Dissipative dynamical systems Part I: General theory. *Archive for Rational Mechanics and Analysis*, **45**(5):321–351, 1972.

[213] J. C. WILLEMS. Dissipative dynamical systems Part II: Linear systems with quadratic supply rates. *Archive for Rational Mechanics and Analysis*, **45**(5):352–393, 1972.

[214] M. N. WIN AND C. D. SMOLKE. Higher-order cellular information processing with synthetic RNA devices. *Science*, **322**:456–460, 2008.

[215] T.-M. Yi, Y. Huang, M. I. Simon, , and J. C. Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *PNAS*, **97**(9):4649–4653, 2000.

[216] Y. Yuan, G.-B. Stan, S. Warnick, and J. Goncalves. Robust dynamical network structure reconstruction. *Automatica*, **47**:1230–1235, 2011.

[217] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1996.

[218] Z. Zi. Sensitivity analysis approaches applied to systems biology models. *IET Systems Biology*, **5**(6):336–346, 2011.

[219] U. Zylstra. Living things as hierarchically organized structures. *Synthese*, **91**:111–133, 1992.