Modeling Frameworks for Modular and Scalable Biological Circuit Design

Thesis by
Ayush Pandey

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended June 21, 2023

Ayush Pandey
ORCID: 0000-0003-3590-4459

ACKNOWLEDGEMENTS

Science is a collective and community effort. This thesis presents contributions from the collaborative efforts of many people and was made possible by the support and guidance of friends and family, who I would like to acknowledge.

I am honored to have Richard Murray as my Ph.D. advisor. Richard's pursuit of engineering large, complex, and robust systems is inspiring and has profoundly influenced me. Everyone who knows Richard knows about his excellent (possibly unmatched) efficiency with everything — emails, meetings, paper reviews, and writing. But, beyond the efficiency and productivity, I have not met anyone in my life who is so objective in decision-making yet so caring about the personal, emotional, and human factors. I truly believe that Richard is unique and the best adviser I could have ever asked for. My words here do not do justice to how grateful I am for his mentoring and strong support over the years. I am sure that I will continue to learn from him in the future.

I am also grateful for the advice and mentoring of my committee members – John Doyle, Niles Pierce, and Domitilla Del Vecchio. In my first year at Caltech, I attended John's classes on networked control systems and robust control theory. To my pleasant surprise, these classes were not just about the topics that the titles might suggest but also about the fundamental theory, architecture, and design of many diverse systems. I joined Caltech to work on communications and control theory. But meetings with John during my rotation project piqued my interest in the math of biology and eventually led me to Richard's group. With Niles and Domitilla, I had the perfect committee members for my research in computational methods and synthetic biology. I have learned about some of the most challenging problems in synthetic biology through collaborations with them. I greatly value the opportunity to work with Domitilla on the context dependence of biological circuits and resource-sharing problems. With Niles, I learned to appreciate the attention to detail and the pursuit of bridging theory to experiments.

In the Biocircuits group, I feel privileged to have worked alongside brilliant scientists and equally kind human beings — a combination that is hard to find, in my opinion. When I joined the lab in 2018, Anandh and Vipul were two senior graduate students working on the theory and modeling of biocircuits. I learned a lot from both as I pivoted my research to the new area of modeling biological systems. I am thankful to both of them for mentoring and boosting my confidence as I navigated unfamiliar terrain. One of my first projects that convinced me of the power of mathematical modeling was the project with Reed McCardell on the modeling of population and composition control of bacterial consortia. The collaboration with Reed was not only fruitful in research output but also a lot of fun. On the same project, I had the pleasure of getting to know and work with Chelsea Hu, who had just joined our group as a postdoc. I learned many unwritten tricks of the trade in mathematical modeling of biological systems from Chelsea. I am deeply thankful for her mentoring over the last five years, not just on research but on everything else in academia. The academic job search would have been much harder if it were not for Chelsea's confidence in me.

In the later years of graduate school, I had the pleasure of collaborating with Zoila Jurado. I greatly admire Zoila's perseverance and innovation in research. Some of my most rewarding research moments come from the students who I mentored over the years. A huge thanks and cheers go out to Makena Rodriguez (Caltech) and Spencer Uyematsu (Pomona College), who just graduated. I learned a lot from their tenacity and dedication. My work with Makena was also published in ACS Synthetic Biology — one of my most proud moments. I am also proud of mentoring Spencer's senior thesis on the mathematical modeling of stem cell differentiation. Apart from those I directly worked with, I am also thankful for the insightful conversations with many colleagues. Cindy, Matt, Liana, Fang, Ankita, Alex, Sam, Rory, Zach, Manisha, Mark, Yan, Elin, John, Mengyi, Apurva, Kellan, Zhewei, Ji, and all other Biocircuits and NCS group members fostered a supportive work environment. A special mention goes to Sumanth, who motivated me to talk to Richard in my first year as I looked for labs. I would also like to acknowledge the two members of the group who ensure everything runs smoothly and are always on top of things — Monica and Miki. Thank you all!

A side quest I have been on is to make all my theoretical work more accessible to scientists by developing associated Python software packages. I was fortunate to have colleagues who shared this vision — William Poole, Zoltan Tuza, Andrey Shur, and Anandh Swaminathan. Thank you for making research more colorful and exciting with BioCRNpyler, Bioscrape, AutoReduce, and Sub-SBML. I deeply cherish all the productive programming sessions with William, which led to papers, workshops, and tutorials at conferences. Other than the biological modeling tools, I had the chance to work on a unique software called Pacti during my last year of graduate school. I am deeply grateful to Inigo for involving me in this project. The research with Inigo has been a highlight of my graduate school, and I cherish our friendship very much. The exceptional diligence and thoroughness with which Inigo addresses research is an inspiration. The research in Chapter 4 and Chapter 5 of this thesis would not have been possible without Inigo and the experimental data that Eric Lei courteously shared with us. Thank you both!

Thanks are also due to my three dear friends: Siddharth Harshe, Ashish Gour, and Utkarsh Tiwari, who have stuck by me for more than a decade now. Utkarsh is officially my cousin, but it feels more appropriate to categorize him as a friend. He also lent a hand with my research during a particularly intense period when I was working non-stop, day and night, to submit my first journal paper. On that note, not just my first paper, but every paper after that, all Python code, job applications, and interviews have been joint work with my partner in research, love, and life, Shailja. My research has greatly benefited from her exceptional problem-solving skills and excellent intuition with theoretical and computational problems. Beyond research, I have found a fun-loving, emotional, and caring partner in Shailja, which, by extension, has made me a better human being as well.

Finally, I would like to dedicate this Ph.D. thesis to my parents. They have struggled and selflessly devoted all their efforts to the best possible education for me. It would not be an overstatement to say that all progress in my life and career is a direct result of the sacrifices, dedication, and hard work of my parents. In a similar vein, my achievements and successes are not just mine to claim — they are shared with my two older brothers as well. This is a collective accomplishment! I am grateful and feel blessed for the infinite love and care that I have been showered with from all members of my family.

ABSTRACT

Synthetic biology is a rapidly evolving interdisciplinary field that combines principles from biology, bioengineering, biochemistry, and computational sciences to design and engineer new biological systems for various applications. This thesis focuses on addressing the challenges in engineering large and complex biological circuits. We develop modular modeling frameworks, formal theory, and computer-assisted design (CAD) tools for design and analysis of biological systems at a larger scale.

This thesis introduces a new problem of robustness in structured model reduction of dynamical systems and provides bounds on a robustness distance metric for linear and nonlinear systems. With this theory, we show the discrimination and quantification of different mathematical models, considering resource loading effects in biological circuits.

Using our proposed model reduction robustness theory and its associated software development, we build a modeling, analysis, and parameter identification pipeline. This pipeline is demonstrated through the characterization of DNA recombination enzymes in a cell-free protein expression system. This pipeline is a general approach to systematically develop mathematical models, infer parameters from experimental data, and guide experimental design choices.

Identification of parameters in detailed mathematical models is a major challenge in synthetic biology where only sparse data is available. This prevents the application of our detail-driven modeling approach to larger biological systems. Hence, to address this limitation, we present a formal methods-based approach for specifying and synthesizing implementations for the design of biological circuits. We present a contract-based design framework for synthetic biology. We write formal description of design objectives at a higher level of abstraction without modeling the details of each component. This design framework facilitates the design and prediction of complex synthetic biological circuits at scale.

Overall, this thesis contributes to the advancement of synthetic biology by providing novel modeling frameworks, analysis methods, and design approaches. These contributions aim to enable the design and analysis of complex biological systems and foster the systematic engineering of biological circuits.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] A. Pandey, M. L. Rodriguez, W. Poole, and R. M. Murray, “Characterization of integrase and excisionase activity in a cell-free protein expression system using a modeling and analysis pipeline,” *ACS Synthetic Biology*, vol. 12, no. 2, pp. 511–523, 2023. DOI: `https://doi.org/10.1021/acssynbio.2c00534`,
Ayush Pandey conceived the project, developed the modeling and analysis pipeline, and conceptualized the experimental design to demonstrate the pipeline. A.P. developed the mathematical models, analyzed the data, and participated in the writing of the manuscript.

[2] A. Pandey*, W. Poole*, A. Swaminathan*, V. Hsiao, and R. M. Murray, “Fast and flexible simulation and parameter estimation for synthetic biology using bioscrape,” *Journal of Open Source Software*, vol. 8, no. 83, p. 5057, 2023. DOI: `https://doi.org/10.21105/joss.05057`,
Ayush Pandey developed the parameter inference and sensitivity analysis features in the Bioscrape software package and applied it to synthetic biology examples. A.P. participated in the software development, standardization, code quality improvements, and in the writing of the manuscript. *denotes equal contribution.

[3] A. Pandey*, I. Incer*, A. Sangiovanni-Vincentelli, and R. M. Murray, “From specification to implementation: Assume-guarantee contracts for synthetic biology,” *International Workshop on Bio-Design Automation (2022), bioRxiv*, pp. 2022–04, 2022. DOI: `https://doi.org/10.1101/2022.04.08.487709`,
Ayush Pandey participated in the conception of the project, applied the assume-guarantee contract theory to engineered biological circuits examples, validation with experimental data, and participated in the writing of the manuscript. *denotes equal contribution.

[4] A. Pandey and R. M. Murray, “Robustness guarantees for structured model reduction of dynamical systems with applications to biomolecular models,” *International Journal of Robust and Nonlinear Control*, vol. 33, no. 9, pp. 5058–5086, 2023. DOI: `https://doi.org/10.1002/rnc.6013`,
Ayush Pandey conceived the project, defined the model reduction robustness problem, derived the theoretical results for linear and nonlinear dynamical systems, applied the theory to biomolecular system examples, and participated in the writing of the manuscript.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Synthetic biology aims to use engineering principles to design and create new biological systems for a variety of applications, such as creating biological sensors, intelligent probiotics, and living materials. For instance, an important problem in healthcare is the lack of complete understanding of various disease mechanisms. Synthetic biology holds the potential to engineer cells that track and comprehend disease-affected metabolites [1], [2]. Current approaches in designing biological circuits rely on tuning of components to work in a specific set of conditions [3], including the host organism [4], growth conditions [5], genetic context [6], and many other factors. As a result, the state of the art in engineering biological circuits is limited to systems of the order of ten engineered components. Additionally, understanding the underlying mechanisms guiding fundamental biological discovery is also limited by scale. In order to design and analyze biological systems at a larger scale, we need modular modeling frameworks, formal theory, and computer-assisted design (CAD) tools. This thesis presents new contributions in all three of these directions. A graphical summary of the contributions is given in Figure 1.1.

Chapter 1 describes synthetic biology examples from engineered bacterial cell populations and cell-free extracts to motivate the need and utility of mathematical modeling. In Chapter 2, we present a new method to build and discriminate between various models for the systems discussed in Chapter 1. With the methodology in Chapter 2 and our development of an iterative parameter inference framework, we apply the modeling tools to models of gene expression in cell-free systems. Finally, we note the limitations of the approach in Chapters 2 and 3 as it is not easily scaled to large system designs (hundreds or thousands of components). Here, we present a contract-based design approach in Chapter 4 for biological circuit design at larger scale. Chapter 4 also touches on synthesis of phenomenological models from specifications written as contracts. An ideal design framework would allow a scientist to go from the specifications (the leftmost panel in Figure 1.1) to detailed implementation models (the rightmost panel in Figure 1.1). We discuss these future directions of research in Chapter 5 of this thesis.

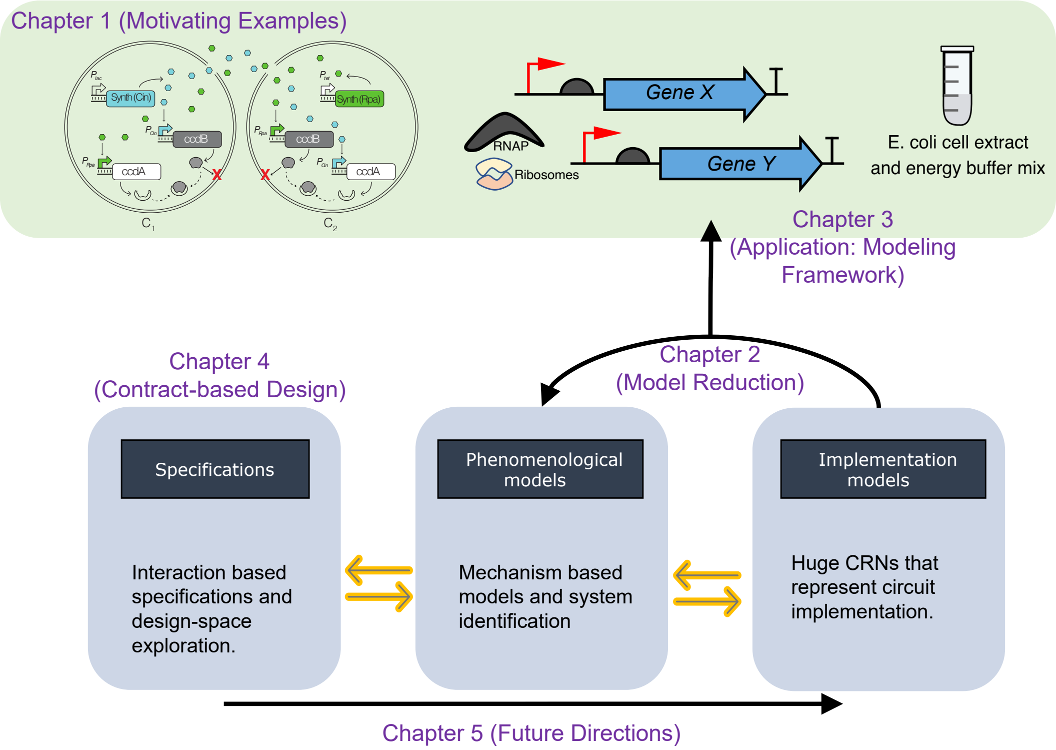


Figure 1.1: An overview of the thesis: Modeling Frameworks for Modular and Scalable Biological Circuit Design.

1.1 Engineering biology

Synthetic biology is an interdisciplinary field that builds on principles of (1) genetics from biology, (2) DNA engineering from bioengineering, (3) chemical synthesis and metabolic engineering from biochemistry, and (4) systematic modeling-based design from computational and mathematical sciences. It is widely accepted in the scientific community that research in synthetic biology has two main classifications: top-down and bottom-up. As the names suggest, the top-down approach entails working with living biological organisms (commonly bacteria [7], yeast [8] or human cells [9]) and changing their genetic components for a particular engineering application. The toggle switch [10] and the repressilator [11] system are two examples of this approach. On the other hand, bottom-up synthetic biology involves starting with individual bio-chemical components such as DNA sequences [12], cell extracts [13], enzymes [14] etc. in order to achieve complex systems with functions that either mimic a biological organism or serve an engineering goal. For the better part of the last two decades, the top-down approach has taken precedence and shown great promise in a wide variety of engineering and health applications [15], [16].

More recently, bottom-up synthetic biology has been gaining some traction as well [17], especially due its modular subsystem design and integration approach. However, a majority of this progress in complex biological circuit design for both top-down and bottom-up synthetic biology has been a result of persistent trial-and-error testing without any clear systematic design approach. In other disciplines such as aerospace engineering, mathematical modeling plays a central role in part and system design. For instance, airplanes are not built by starting with big metal pieces and trying out combinations or exploring the full design space but rather emerge as a result of careful characterization and integration of smaller parts together. One could even argue that this systematic approach is one of the key reasons for the pivotal success of electrical, aerospace, or mechanical engineering. We would like to emulate this success for biological systems and bioengineering applications by developing modeling frameworks and system integration design approaches that are generally applicable to a large class of engineered biological systems.

1.2 Modeling framework for biological circuits

Mathematical modeling has been extensively used in the design of engineering systems [18]. Synthesis methods for linearized approximations of subsystems have shown great promise in the design process. The comprehensively developed theory of linear algebra, feedback control, and analysis has aided in the unimpeded growth of modeling based design of engineering systems. Nonlinear analysis has also played its role in simulating complicated processes such as in fluids and solid dynamics [19]. However, the extension of linear control design and nonlinear analysis approaches has not quite made its mark on synthetic biology yet. The reasons for this are diverse, although a common theme is that generalization of results across different biological systems is hard due to strong context-dependence. As a result, researchers have only been able to use particular mathematical models to describe various biological effects to aid with system design, such as resource loading [20], stochasticity [21], noise [22], and host-cell effects [23]. The state of the art in engineering biological circuits is limited by scale where some of the most complex systems that have been engineered [24], [25] have order of ten engineered components. Similarly, understanding the underlying mechanisms guiding fundamental biological discovery is also limited by scale. In order to design and analyze biological systems at a larger scale, we need modeling frameworks and computer-assisted design (CAD) tools. In this thesis, we build these standardized software for the modeling and analysis of biological systems.

In a collaborative effort, we have developed a unique biological model compilation Python package called BioCRNpyler [26]. This tool can quickly build large biological system models from simple high-level specifications. For example, a gene regulatory network model with hundreds of species and reactions can be compiled in less than ten lines of Python code with an option to tune the granularity of the modeling details. A similar software tool called OneModel [27] has been developed recently which can also build differential-algebraic equation models. However, a few key limitations of these software limit their accessibility and compatibility for building models of systems. On the other hand, modeling tools such as iBioSim [28], Tellurium [29], and COPASI [30] provide a much more accessible framework for building biological system models but cannot be scaled to build large system models easily.

Training mathematical models with biological data is an integral step in any modeling effort. Formal systems design methods in engineering make use of mathematical models and data from characterized components and their environments. To imitate this for biological system analysis, it is crucial to integrate experimental data with complex biological system models. In this thesis, we extend the biological simulation software package called Bioscrape [31] to implement easy-to-use Bayesian inference methods. However, Bioscrape only supports time-series data and only a few built-in simulation and analysis tools. For wider adoption of the system analysis and identification tools in Bioscrape, it is vital to develop software that can support multiple data modalities.

For the modeling tools described above, the user implicitly chooses the detail of the models and the underlying assumptions. In this way, all mathematical models are “reduced models” where the modeling details are decided depending on the hypothesis under study. Reduced-order modeling is ubiquitous in all engineering applications. Controller synthesis or system design makes considerable use of reduced-order implementations in many applications. Abstracting away the details of a system model to focus on modeling the properties of interest and its interactions is an important insight that is commonly used in control systems’ design. Reduced models are also useful to specify the desired objectives or the performance specifications of a system. To meet these objectives, the designer needs to map the specifications to the mechanistic implementations and also mathematically characterize this mapping [32]–[34] in order to understand and analyze the system performance. However, the current modeling paradigm in synthetic biology does not explicitly account for conditions when the system operates in situations where the underlying assumptions are not satisfied.

In this thesis, we present a new model reduction method and associated software tool called AutoReduce. We also further develop on BioCRNpyler (a detailed chemical reaction modeling software) and Bioscrape (a biological model simulator and inference toolbox). With these, we create a modeling framework that can be used to systematically build mathematical models of tunable granularity, assess the goodness of models, and use these tools in an iterative pipeline to guide larger system designs.

1.3 Motivating examples

We present three motivating examples from synthetic biology where system modeling, simulation, and analysis have led to useful results. We also discuss some of the limitations with these approaches and highlight the advances that this thesis makes in bridging the existing gaps.

Wound healing

Synthetic biology can be used to engineer efficient drugs [1], metabolic pathways [35], and track cell populations [2] for precision medicine. Robust and dynamical sensing of the disease biomarkers could provide an exciting perspective on designing diagnostic tools for treatment. For example, the process of wound healing in humans involves complex control of many different kinds of cellular consortia in tandem. To use bacterial cells as a proof-of-concept to demonstrate and control the process of wound healing, we need to engineer cells such that we can precisely control the population of each cell and the ratio of a particular type of cell with others.

We built a phenomenological ordinary differential equation model of an engineered two-member bacterial community to show that we can control the density and composition of the bacterial consortia [7]. In this system model, we have two different cell types (C_1 and C_2). Each cell type expresses a toxin that kills itself but at the same time activates an anti-toxin that rescues the other cell type. In this way, we expect the system to demonstrate population and ratiometric control.

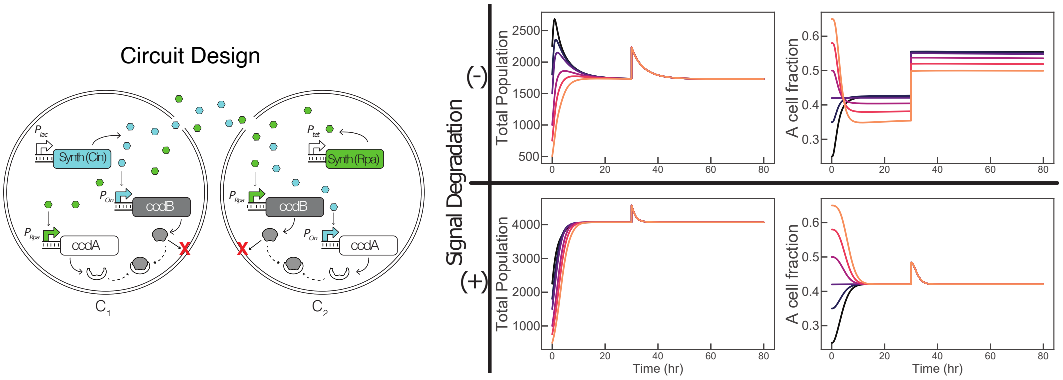


Figure 1.2: A part of the wound healing system design where *in vivo* circuits are designed to control the population of both cell types A and B while also maintaining a desired ratio of the cells. Plots are simulations of cell density (total population) and composition (cell fraction). Each colored curve represents seeding a community at a unique composition and density. Without intercellular signal degradation, we observe that the density converges to a steady state while composition does not converge. But, with signal degradation we observe that the disturbances are rejected as the consortium maintains the population and the composition.

In our first iteration of the model simulations, although the population was robustly controlled to the desired fixed point, we observed that the composition control was not robust to disturbances introduced in the system. By analyzing the stability of the system, we predicted that active degradation of quorum sensing signals mediated by the AiiA degradase enzymes robustly adapts to a desired set-state for population of each cell and the desired composition [7]. The biological circuit that we studied is shown in Figure 1.2 and the associated simulations of the ODE model that we developed are also shown.

Engineering disturbance rejection in bacteria

Similar to the previous motivating example on wound healing and population control, modeling and simulations can also be used to study layered feedback control circuits [36]. In this system, a combination of a slow-but-accurate and fast-but-inaccurate feedback is used to achieve precise control of protein expression even in the presence of environmental disturbances.

Feedback control theory using linearization of system model is used to validate the performance of the systems *in silico*. Further exploration using the nonlinear model is used to explain the observed system behavior. This is another example that

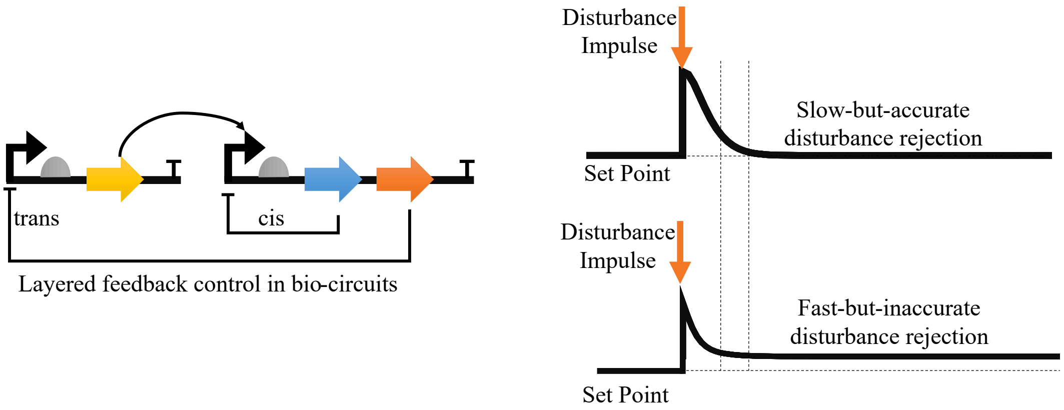


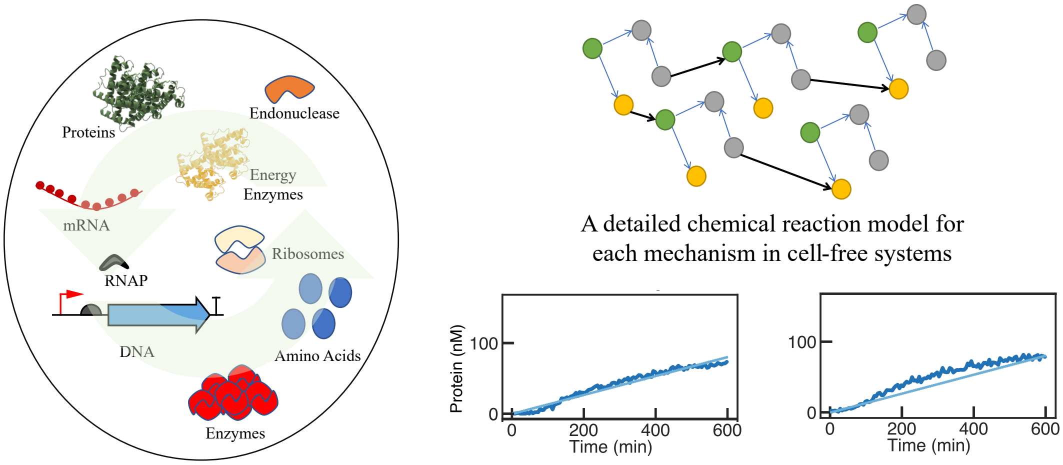
Figure 1.3: Feedback mechanisms can be engineered to explore fundamental design tradeoffs in biological circuit design [36]. A disturbance may be rejected with the use of negative feedback (cis) that fully recovers to the desired steady-state slowly. On the other hand, a trans negative feedback may be engineered that does not fully recover to the desired set point but rejects the disturbance quickly. The two different feedback mechanisms are shown as repression arrows in the circuit diagram.

highlights the importance of phenomenological models for engineered biological systems. But this may not always be the case, as shown with the next example.

Optimizing artificial cell systems

Cell-free synthetic biology [37] and artifical cell-like system design [38] are burgeoning research areas within synthetic biology. With cell-free systems, we hope to engineer larger biological circuits without strong context-dependence that comes from complex processes of living cells. However, batch-to-batch variability of cell-free systems and their short “runtime” are major drawbacks. To address these issues, mathematical models could be developed that reliably predict the protein expression in a cell-free system. In particular, for cell-free systems like PURE [39], all components of the mixture are known, so we may expect that a detailed mechanistic model can capture all possible interactions. This has the potential to accurately predict the performance of cell-free based circuits and artificial cell systems.

The motivating examples above highlight the vast possibilities where mathematical modeling can play a crucial role and point out some of the gaps in the current modeling frameworks and design tools. In Chapter 2, we present the theory of assessing robustness of reduced-order models. Using this theory, we develop a modeling framework that can be used to certify the goodness of different models. We apply this framework to model biological systems presented here as motivating



A detailed chemical reaction model for each mechanism in cell-free systems

Cell-free protein expression system

Validated models that predict protein expression

Figure 1.4: The role of mathematical modeling in predicting protein expression in cell-free protein expression systems. Due to the relatively lower complexity of mechanisms in cell-free systems as compared with living cells, detailed chemical reaction network models may be built that would accurately predict the protein expression.

examples — negative feedback in robustly controlling bacterial protein expression, and gene expression models. We expand on the gene expression models in the cell-free protein expression system context in Chapter 3. In this chapter, we also present a modeling and analysis pipeline that is generally applicable to engineered biological systems. Finally, in Chapter 4 and Chapter 5, we present a new theoretical framework based on assume-guarantee contracts for predictive and compositional system modeling.

Chapter 2

BUILDING BIOLOGICAL SYSTEM MODELS

2.1 Introduction

We consider the problem of systematically building reduced biological system models from detailed system descriptions. We derive results on the robustness of the error in structured model reduction for both linear and nonlinear dynamical systems. The robustness of structured model reduction is a well-studied problem for singularly perturbed dynamical systems. However, singular perturbation theory requires the system dynamics to be in time-scale separable form which is not generally possible for physical system models. We present a more general robustness analysis approach that does not require the system to be in the standard singular perturbation form. Moreover, with our method, we give robustness guarantees to each reduced model with respect to parametric uncertainties in all model parameters.

In this chapter, we define a new problem for the robustness of error in structured model reduction of dynamical systems. We derived bounds on a robustness distance metric for linear and nonlinear dynamical systems using two different approaches. The linear analysis approach quantifies the robustness in model reduction that depends on the Lyapunov matrix and the parameter derivative of the system matrix. For nonlinear dynamical systems, we develop a sensitivity analysis-based approach to elucidate the robustness analysis similar to the linear analysis approach. We also prove that for linear systems both methods lead to equivalent results.

Our novel problem setup of estimating robustness in model reduction leads to interesting new results in the mathematical modeling of a gene expression system. We derive a new mathematical model that can capture the effects of resource loading in gene expression. Under different biologically plausible conditions such as weak ribosome binding strength, RNA polymerase loading, our approach can computationally discriminate between different possible reduced models and quantify its performance guarantees.

Chemical reaction network modeling

A chemical reaction network (CRN) is a set of species and chemical reactions where each reaction is associated with a rate function or propensity that models the changes in the species' concentrations or counts. Under certain assumptions, CRNs can be written as ordinary differential equations (ODEs) and then numerically integrated [40]. CRNs can also be simulated stochastically by using the reactions to generate continuous-time Markov chains. [41] or by writing a chemical master equation [42]. CRNs are prevalent in biological system modeling as they can represent systems in a mechanistic manner and provide flexibility in the choice of simulation frameworks. CRN theory is an extensively studied topic with applications in various sub-fields such as mathematical modeling [43], computer science [44], and thermodynamics [45]. Despite the strong theoretical underpinnings of CRN modeling, many biological system models in practice are empirical and heuristic-driven. This is due to the lack of information about the possible chemical reactions in a system and the even harder problem of estimating the reaction propensities given the sparse experimental evidence. We classify these models as phenomenological in nature and discuss them next.

Phenomenological modeling

The lack of mechanistic details of various biological processes prompts us toward higher abstractions and model the phenomena of the biological system in question. Such phenomenological models are often easier to fit the observed experimental data, but lack the mechanistic details of the system. More importantly, the correctness of phenomenological models is dependent on many implicit assumptions. It is important to monitor these assumptions when using these models in practice to avoid incorrect conclusions from the model under conditions that violate the underlying assumptions. The task of constructing correct models is made more challenging by the difficulty in discerning between correct and incorrect models using experimental data [46]–[48]. Hence, it is important to construct system specifications that guarantee certain sets of behaviors under explicitly stated assumptions. Such models can be used to reject incorrect models based on unsatisfiability of the assumptions or the guarantees. We discuss such models as specifications in detail in Chapter 4. In this chapter, we explore the mapping between the detailed models (for example, CRNs) and their reduced-order representations (towards phenomenological models). Using this mapping, we provide a computational framework to systematically build biological system models under given mechanistic assumptions about the system.

Reduced-order modeling

For applications of control theory to physical system design, a reduced model is commonly used that describes the dynamics of interest in lower dimensions to simplify the design process. Reduced models are used to specify the desired objectives or the performance specifications of a system. To meet these objectives, the designer needs to map the reduced models to the level of system design and also mathematically characterize this mapping to understand and analyze system performance. For biological systems, this is a challenge that hinders the use of mathematical models in experimental designs and analysis to some extent.

For model-based design of biological circuits, we need to develop mathematical models that map system design specifications to mechanistic details. Commonly used phenomenological models are based on empirical information and their parameters describe lumped properties of the system that are effective in explaining the observed experimental data [7], [10], [11], [31], [49] by solving the inverse problem [50] but have not been readily used for forward engineering of biological circuits [51]. Towards that end, to explore different design possibilities one needs to carefully justify the validity of the underlying assumptions for each model [52]. Time-scale separation is one of the most common properties exhibited by biological systems. For example, the half-life of an mRNA in *E. coli* is around 100 seconds whereas the average half-life for a protein is of the order of a few hours. Hence, it is important to consider the effects of this assumption on biological system models.

Structured model reduction

Singular perturbation theory [53] is the formal way of deriving mathematical models for system dynamics with time-scale separation. For such systems, it is common to separate the dynamics into fast and slow modes. Then, singular perturbation theory can be used to derive reduced models that accurately represent the dynamics of interest. A key feature of singular perturbation theory is that the states of the reduced model are a subset of the states of the full model. In other words, the structure of the model and the meaning of the states and parameters is conserved by construction in any reduced model obtained using singular perturbation theory. This is not automatically the case in other model reduction techniques where transformations are introduced [54], [55] and hence in such techniques, the meanings of the states may not be preserved.

We define structured model reduction as the set of model reduction methods where the states of the reduced models are a strict subset of the states of the full model. The advantage of structured model reduction techniques is that an explicit mapping between the full and the reduced model is readily available [34]. Moreover, since the parameters and the states in the reduced model have the same meaning as in the full model, the design outputs and analysis results obtained using the reduced model can easily be given context and compared with the full model [56]. This is especially relevant for biological system modeling as it is often important to map the reduced dynamics to the mechanistic details. However, due to the strict condition on the possible reduced model states, structured model reduction methods suffer from the limitation that the choice of reduced models is restricted and dependent on the modeling details of the full system. That is, for a given full model it may not always be possible to analytically derive a reduced model. Other model reduction methods that are projection-based or those which preserve the input-output mapping are better in that respect. Here, we focus on the former class of model reduction problems that preserve the modeling structure in the reduced models and hence more relevant for biological systems.

Metrics for model reduction

The goal with any model reduction problem is to minimize the error in the performance of the reduced model when compared to the full model. This error performance criterion can be general and depend on trajectories of all states and output variables, or specific, such as minimizing a particular metric of interest. Singular perturbation theory for model reduction and its error analysis is a widely studied topic in the literature [57]–[59] for different system and control design settings. A commonly used method for model reduction that is derived from the singular perturbation concept of time-scale separation is the quasi-steady-state approximation (QSSA) method [60]–[63]. Here, a subset of states is assumed to be at steady-state and hence their dynamics are collapsed to algebraic relationships. Error analysis for QSSA based model reduction [64]–[66] has been studied as well. However, robustness of these model reduction methods is not as widely studied in the literature.

Robust control design is a well-studied problem in control theory. The extensions of robust control theory to singularly-perturbed systems are studied in [67] and [68]. Similarly, robust stability analysis of adaptive control problems, linear time-varying systems, and the general parametric uncertainty problems has been of interest as well [69], [70]. A complementary, although not as widely applicable, approach to study the robustness of systems is to use sensitivity analysis of system variables or derived properties under parameter variations. Due to the success of robust control design methods [69] for different applications, the more holistic approach of sensitivity analysis for robustness estimates has not received much attention. In [71], sensitivities of singular values are used to give estimates for robustness properties of a linear feedback system. The advantage of such a method is that it analyzes the effect of multiple parametric uncertainties and hence can be used to enhance the usual robust stability approaches.

Our problem statement is motivated by this sensitivity analysis approach for robustness and by the lack of existing results for robustness estimates of error in structured model reduction. In particular, we give robustness guarantees for the error in model reduction under parametric uncertainties. Using linear analysis, we give a robustness metric for QSSA-style model reduction of linear dynamical systems. We present a complementary approach that employs sensitivity analysis of the error in model reduction to estimate the robustness under parametric uncertainties. This approach works for nonlinear dynamical systems as well. We demonstrate our method with the help of common biomolecular system examples. Using our results on the model reduction robustness metric, we derive a new mathematical model for gene expression that accounts for ribosomal loading better than the commonly used models in the literature.

2.2 Mathematical preliminaries

We denote an eigenvalue of a matrix P by $\lambda(P)$. The maximum eigenvalue will be denoted by $\lambda_{\max}(P)$. For a state-dependent matrix $P(x)$ we denote the maximum eigenvalue of P over all values of x by $\lambda_{\max_x}(P)$. We consider the Euclidean 2-norm for vectors. For example, we use the notation $||x||$ for the 2-norm of $x \in \mathbb{R}^n$ and similarly for matrices $||\cdot||$ represents the induced 2-norm.

Problem formulation

We start by formulating the problem for linear system dynamics and then focus on the nonlinear systems next. The full linear system model is given by the following autonomous state-space equation

$$\dot{x} = A(\theta)x, \quad y = Cx, \quad x(0) = x_0, \tag{2.1}$$

where $x \in \mathbb{R}^n$ are state variables, the output vector is $y \in \mathbb{R}^k$, and $\theta = \left[\theta_1, \theta_2, ..., \theta_p\right]^T$ is the vector of all model parameters. We consider a structured model reduction where the dynamics of a subset of states (x_c) are collapsed (converted to algebraic relationships) on account of being at quasi-steady-state. The remaining states are the states of the reduced model, $\hat{x}$. The reduced model is given by

$$\dot{\hat{x}} = \hat{A}(\theta)\hat{x}, \quad \hat{y} = \hat{C}\hat{x}, \quad \hat{x}(0) = \hat{x}_0, \tag{2.2}$$

where $\hat{x} \in \mathbb{R}^{\hat{n}}$ are the reduced state variables and $\hat{y} \in \mathbb{R}^k$ is the output vector. We assume that the full and the reduced model have the same number of outputs but different dynamics. Throughout this chapter, we also assume that both the full and the reduced systems are asymptotically stable and observable. This model reduction corresponds to a time-scale separation in the dynamics of the full model where the meaning of all states and parameters is retained in the reduced model. It is a relaxed form of singular perturbation theory based model reduction in that it does not require the system to be in the standard separable form, which is a hard condition to satisfy for general system dynamics, as we will see next.

Singular perturbation theory [53] is the standard way to derive reduced models and bounds on error in model reduction for the problem statement given above. However, to use singular perturbation theory the system dynamics need to be separable according to the different time scales. For the problem formulation above, the requirement would be that we can write the dynamics in the following form,

$$\begin{aligned} \dot{\hat{x}} &= A_{11}\hat{x} + A_{12}x_c, \\ \epsilon\dot{x}_c &= A_{21}\hat{x} + A_{22}x_c, \end{aligned} \tag{2.3}$$

where $A_{11} \in \mathbb{R}^{\hat{n}\times\hat{n}}$, $A_{22} \in \mathbb{R}^{(n-\hat{n})\times(n-\hat{n})}$, and similarly we have A_{12} and A_{21}. Now, under the condition that $\epsilon \longrightarrow 0$ and Hurwitz A_{22}, we get the reduced model as in equation (2.2) with,

$$\dot{\hat{x}} = (A_{11} - A_{12}A_{22}^{-1}A_{21})\hat{x} := \hat{A}\hat{x}. \tag{2.4}$$

In Figure 2.1, the block diagram for a system with time-scale separation is given. Using singular perturbation theory, conditions can be derived under which the error in model reduction converges to zero:

$$||x - \hat{x}|| \leq O(\epsilon) \tag{2.5}$$

for a time-scale separation parameter $\epsilon \in [0, \epsilon^*]$. The error in model reduction goes to zero as $\epsilon \longrightarrow 0$. This is the standard model reduction problem using singular

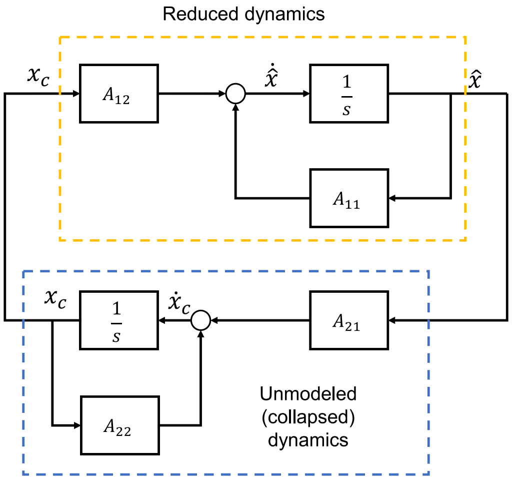


Figure 2.1: Structured model reduction of a linear system. The $1/s$ block represents an integrator.

perturbation to separate the time scales of the model where the dynamics of the "fast" states of the system (x_c) are collapsed to zero when $\epsilon \longrightarrow 0$ and the dynamics of the "slow" states ($\hat{x}$) is the reduced dynamics.

Under uncertainties in the system dynamics, it is important to analyze the robustness of the model reduction. The results on robustness for singular perturbation based model reduction either focus on robust controller design under uncertainties for singularly perturbed systems [67], [68] or analyze the effect on the time-scale separation parameter ϵ due to uncertainties. Recently, a singular perturbation margin [72] (similar to the gain margin and phase margin definitions) has been proposed to assess the robust stability of singularly perturbed systems under uncertainties. It is defined using the ϵ^* given in the error approximation equation (2.5). This framework can be used to compute a robustness estimate of the model reduction error for the singular perturbation method. An extension of this robust stability margin for nonlinear dynamics is given in [73]. Despite the rich body of literature on singular perturbation theory, the major limitation of such a model reduction approach remains that the system dynamics must be written in the standard form (2.3). As stated in [74], for physical systems it is usually not straightforward to put a model in the singularly perturbed form since the choice of combination(s) of parameters to be considered small is not always clear. Hence, the relaxed approach of using QSSA is common for various applications.

In QSSA, the dynamics of a set of states are collapsed to zero to get the reduced-order model. The choice of states to be collapsed is usually driven by known time-scale separations in the system model. Although QSSA is a widely used approach it does not necessarily guarantee error performance as in equation (2.5). The limitation of QSSA is that the mathematical justification and conditions for approximating a variable to be at a steady-state are not always obvious. As a result, there could be many possibilities of reduced models and so it is the designer's task to find a "correct" QSSA based model reduction. Towards that end, in [66], a structured model reduction algorithm is presented that guides the choice of collapsed states so that the error between the output of the full and the reduced models is minimized. Other QSSA error analysis [64] approaches can also be used for this purpose. In this chapter, we briefly discuss model reduction error results for QSSA further. Our main focus is on the problem of robustness of this structured model reduction, that is, how robust a particular model reduction is under parametric uncertainties.

To formulate this problem, we first construct an augmented state-space system that consists of variables of the full and the reduced model together:

$$\bar{x} := \begin{bmatrix} x \\ \hat{x} \end{bmatrix}.$$

We denote all augmented variables similarly with a bar on top of the usual variables. So,

$$\bar{A} := \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix}.$$

For the augmented state variable, we can write the following state-space system,

$$\dot{\bar{x}} = \bar{A}(\theta)\bar{x}, \quad \zeta = \bar{C}\bar{x}, \qquad \bar{x}(0) = \begin{bmatrix} x_0 \\ \hat{x}_0 \end{bmatrix}, \tag{2.6}$$

where ζ is the error in model reduction defined as $\zeta = y - \hat{y}$, hence $\bar{C} = \begin{bmatrix} C & -\hat{C} \end{bmatrix}$.

To study the robustness of the structured model reduction (that is the robustness of deriving the particular reduced model $(\hat{x}, \hat{A})$ under uncertainties in model parameters) we need an upper bound on $\|\zeta\|$ as the parameters θ vary. For the linear augmented system we can write the following by solving for $\zeta(t, \theta)$,

$$\zeta(t, \theta) = \bar{C}e^{\bar{A}t}\bar{x}(0), \tag{2.7}$$

where $\zeta \in \mathbb{R}^k$.

Lemma 1 (See [75])**.** *For a Hurwitz matrix A, the norm of the matrix exponential is bounded above as*

$$\left\|e^{At}\right\| \le e^{-|\mu|t}$$

for all $t \ge 0$, where μ is the logarithm norm of A [76]. For the log-norm induced by the 2-norm, we have that

$$\mu(A) = \frac{\lambda_{\max}(A + A^T)}{2},$$

and for Hurwitz A, μ is always negative.

We use this to give an important result on the derivative of the matrix exponential with respect to a parameter.

Lemma 2. *The derivative of the matrix exponential e^{At} with respect to a parameter θ_i is given by*

$$\frac{\partial e^{At}}{\partial \theta_i} = \int_0^t e^{(t-\tau)A} \frac{\partial A}{\partial \theta_i} e^{\tau A} d\tau. \tag{2.8}$$

If A is Hurwitz, the norm of the derivative of the matrix exponential with θ_i is bounded above by

$$\left\|\frac{\partial e^{At}}{\partial \theta_i}\right\| \le \left\|\frac{\partial A}{\partial \theta}\right\| t e^{-|\mu|t} \tag{2.9}$$

where $|\mu|$ is the absolute value of the log-norm of A as in Lemma 1.

Proof. The first part of the lemma (in equation (2.8)) is a result proven in [77] and a simplified version is given in [78] and [79]. An alternate proof for equation (2.8) is presented here. For a linear system, $\dot{x} = Ax$, we can write the solution $x(t) = e^{At}x(0)$, where e^{At} is the matrix exponential. Now, for a parameter θ_i, we can write

$$\frac{\partial x(t)}{\partial \theta_i} = e^{At}\frac{\partial x(0)}{\partial \theta_i} + \frac{\partial e^{At}}{\partial \theta_i} x(0),$$

using the product rule of differentiation. Define

$$S(t) := \frac{\partial x(t)}{\partial \theta_i}, \tag{2.10}$$

so we have,

$$S(t) = e^{At}S(0) + \frac{\partial e^{At}}{\partial \theta_i} x(0), \tag{2.11}$$

and write $\dot{S}(t)$ using equation (2.10) and $\dot{x} = Ax$ as

$$\frac{dS}{dt} = A\frac{\partial x}{\partial \theta_i} + \frac{\partial A}{\partial \theta_i} x = AS + \frac{\partial A}{\partial \theta_i} x.$$

Solving for $S(t)$, we get,

$$S(t) = e^{At}S(0) + \int_0^t e^{A(t-\tau)}\frac{\partial A}{\partial \theta_i}x(\tau)d\tau.$$

Since $x(\tau) = e^{A\tau}x(0)$, we can simplify the above equation and write

$$S(t) = e^{At}S(0) + \left[\int_o^t e^{A(t-\tau)}\frac{\partial A}{\partial \theta_i}e^{A\tau}d\tau\right]x(0).$$

Comparing this with equation (2.11), we get the desired result for the derivative of the matrix exponential

$$\frac{\partial e^{At}}{\partial \theta_i} = \int_o^t e^{A(t-\tau)}\frac{\partial A}{\partial \theta_i}e^{A\tau}d\tau.$$

To prove the second part, given in equation (2.9), write the norm of the derivative of the matrix exponential with respect to a parameter θ_i as

$$\left\|\frac{\partial e^{At}}{\partial \theta_i}\right\| = \left\|\int_0^t e^{(t-\tau)A}\frac{\partial A}{\partial \theta_i}e^{\tau A}d\tau\right\| \leq \int_0^t \left\|e^{(t-\tau)A}\right\|\left\|\frac{\partial A}{\partial \theta_i}\right\|\left\|e^{\tau A}\right\| d\tau.$$

Now using the result of Lemma 1, since A is Hurwitz, we have,

$$\left\|\frac{\partial e^{At}}{\partial \theta_i}\right\| \leq \left\|\frac{\partial A}{\partial \theta_i}\right\|\int_0^t e^{-(t-\tau)|\mu|}e^{-\tau|\mu|}d\tau.$$

Solving the above integral, we get the desired result:

$$\left\|\frac{\partial e^{At}}{\partial \theta_i}\right\| \leq \left\|\frac{\partial A}{\partial \theta_i}\right\| te^{-|\mu|t}. \qquad \square$$

Note that under our assumption of asymptotically stable full and reduced models, the augmented block-diagonal matrix $\bar{A}$ is Hurwitz. So, with the result from Lemma 1, we can conclude that the error dynamics given in equation (2.7) converges to zero at steady-state. In this way, we have set up the problem to focus solely on the analysis of the robustness of model reduction while assuming that the problem of minimizing the model reduction error has already been addressed. For a given structured model reduction (and hence the corresponding augmented system above), we can get a bound on the error in model reduction as the model parameters vary to give a robustness estimate for this model reduction. We construct a normalized [80] robustness distance estimate for this purpose by computing the change in error with parameter perturbations around any nominal values given by θ_i^*:

$$d_R = \sum_{i=1}^{p} \frac{\theta_i^*}{\left\|\zeta(t, \theta_i^*)\right\|} \cdot \left\| \left.\frac{\partial \zeta}{\partial \theta_i}\right|_{\theta_i = \theta_i^*} \right\|, \tag{2.12}$$

where $\zeta(t, \theta_i^*)$ is the non-zero error in model reduction for $t > 0$ and parameter $\theta_i = \theta_i^*$. We define the sensitivity of the error with respect to a parameter θ_i in the equation above as S_ζ:

$$S_\zeta = \frac{\partial \zeta}{\partial \theta_i},$$

where $S_\zeta \in \mathbb{R}^k$. Using this distance estimate, we propose the following robustness metric to determine the performance of reduced models under parameter uncertainties:

$$R = \frac{1}{1 + d_R}. \tag{2.13}$$

Clearly, for any reduced model we have that $R \in (0, 1)$ with $R \longrightarrow 0$ corresponding to worst robust performance and $R \longrightarrow 1$ corresponding to the best robust performance. Hence, our goal is to compute R (through a bound on the norm of S_ζ) to give a robustness metric for each possible reduced model. The results in the next section give upper bounds to the norm of S_ζ for linear and nonlinear system settings. With the help of some examples, we then demonstrate the computation of the robustness metric as discussed above using these bounds on $\left\|S_\zeta\right\|$. Finally, to decide a particular reduced model we may use a combination of the error and the robustness metrics to choose a particular reduced model in a given parameter regime.

2.3 Robustness of reduced models

Linear system — Uncertain initial conditions

In this section, we consider the uncertainties in the initial conditions — $x(0)$ of a linear system. We can use this result to assess the robust performance of different possible structured model reductions when the initial conditions are dependent on the uncertain model parameters.

Theorem 1. *For the structured model reduction of the autonomous linear system (2.1) to the reduced form of system (2.2) by using time-scale separation and quasi-steady-state approximation under uncertain initial conditions, the norm of the sensitivity of the error in model reduction S_ζ is bounded above by*

$$\left\|S_\zeta\right\|_2^2 \leq \lambda_{\max}(P) \left\|\frac{\partial \bar{x}(0)}{\partial \theta_i}\right\|_2^2,$$

where P is the Lyapunov matrix that solves the equation $\bar{A}^T P + P\bar{A} = -\bar{C}^T\bar{C}$.

Proof. We have the 2-norm [69] of S_ζ defined as:

$$\left\|S_\zeta\right\|_2^2 = \int_0^\infty S_\zeta(t)^T S_\zeta(t) dt.$$

Using equation (2.7), we can write S_ζ for a parameter θ_i as

$$S_\zeta = \bar{C}e^{\bar{A}t}\frac{\partial \bar{x}(0)}{\partial \theta_i},$$

since we have assumed that the matrices $\bar{A}$ and $\bar{C}$ are not dependent on parameters. The norm of S_ζ then becomes

$$\|S_\zeta\|_2^2 = \int_0^\infty \left(\frac{\partial \bar{x}(0)}{\partial \theta_i}\right)^T e^{\bar{A}^T t}\bar{C}^T\bar{C}e^{\bar{A}t}\left(\frac{\partial \bar{x}(0)}{\partial \theta_i}\right)dt. \tag{2.14}$$

From [81, Ch.5], we know that for an observable asymptotically stable system, there exists a unique matrix P that solves the Lyapunov equation $\bar{A}^T P + P\bar{A} = -\bar{C}^T\bar{C}$ given by the observability Gramian:

$$P = \lim_{N\to\infty} W_{\mathrm{o}}(N) = \lim_{N\to\infty}\int_0^N e^{\bar{A}^T t}\bar{C}^T\bar{C}e^{\bar{A}t}dt,$$

where $W_{\mathrm{o}}(N)$ is the observability Gramian. Substituting this into equation (2.14) gives us the desired result:

$$\|S_\zeta\|_2^2 = \left(\frac{\partial \bar{x}(0)}{\partial \theta_i}\right)^T \left[\int_0^\infty e^{\bar{A}^T t}\bar{C}^T\bar{C}e^{\bar{A}t}dt\right]\left(\frac{\partial \bar{x}(0)}{\partial \theta_i}\right)$$

$$\Rightarrow \|S_\zeta\|_2^2 \le \lambda_{\max}(P)\left\|\frac{\partial \bar{x}(0)}{\partial \theta_i}\right\|_2^2. \ \square$$

Linear system — Uncertain system dynamics

Now we consider the case where the system dynamics given by the $\bar{A}(\theta)$ matrix is dependent on uncertain parameters. For simplicity we denote $\bar{A}(\theta) := \bar{A}$, noting that it is parameter-dependent.

Theorem 2. *For the structured model reduction of the autonomous linear system (2.1) to the reduced form of system (2.2) by using time-scale separation and quasi-steady-state approximation under uncertain system dynamics, the norm of the sensitivity of the error in model reduction S_ζ is bounded above by*

$$\|S_\zeta\|_2^2 \le \tilde{M}\left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|_2^2 \|\bar{C}^T\bar{C}\|_2 \|\bar{x}(0)\|_2^2,$$

where $\tilde{M} = 1/4\,|\mu|^3$ and μ is dependent on $\bar{A}$ as given in Lemma 1.

Proof. Write the norm of S_ζ as

$$\|S_\zeta\|_2^2 = \int_0^\infty S_\zeta(t)^T S_\zeta(t) dt. \tag{2.15}$$

To derive the bounds, we first write the partial derivative of $\zeta(t, \theta)$ with respect to a parameter θ_i as given in equation (2.7),

$$S_\zeta = \frac{\partial \zeta}{\partial \theta_i} = \bar{C} \frac{\partial e^{\bar{A}t}}{\partial \theta_i} \bar{x}(0), \tag{2.16}$$

assuming that the output matrix $\bar{C}$ and the initial conditions are independent of model parameters. We can write the norm of S_ζ as,

$$\|S_\zeta\|^2 = \int_0^\infty \bar{x}(0)^T \left(\frac{\partial e^{\bar{A}t}}{\partial \theta_i}\right)^T \bar{C}^T \bar{C} \left(\frac{\partial e^{\bar{A}t}}{\partial \theta_i}\right) \bar{x}(0) dt \leq \int_0^\infty \left\|\frac{\partial e^{\bar{A}t}}{\partial \theta_i}\right\|^2 \left\|\bar{C}^T \bar{C}\right\| \|\bar{x}(0)\|^2 dt.$$

Using the result from Lemma 2, we can write,

$$\|S_\zeta\|^2 \leq \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|^2 \left\|\bar{C}^T \bar{C}\right\| \|\bar{x}(0)\|^2 \int_0^\infty t^2 e^{-2|\mu|t} dt.$$

We can evaluate the integral above by parts:

$$\int_0^\infty t^2 e^{-2|\mu|t} dt = \frac{1}{4|\mu|^3}.$$

We get the desired result for the norm of S_ζ by substituting this integral,

$$\|S_\zeta\|^2 \leq \frac{1}{4|\mu|^3} \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|^2 \left\|\bar{C}^T \bar{C}\right\| \|\bar{x}(0)\|^2.$$

The robustness metric R follows by using the above bound and equation (2.13). □

Corollary 2.1. *Under simultaneous parametric uncertainties in system dynamics and initial conditions, we can write the norm of S_ζ as*

$$\begin{aligned} \|S_\zeta\|_2^2 \leq \lambda_{\max}(P) \left\|\frac{\partial \bar{x}(0)}{\partial \theta_i}\right\|_2^2 &+ \frac{1}{4|\mu|^3} \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|_2^2 \left\|\bar{C}^T \bar{C}\right\|_2 \|\bar{x}(0)\|_2^2 \\ &+ \frac{1}{2|\mu|^2} \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|_2 \left\|\bar{C}^T \bar{C}\right\|_2 \|\bar{x}(0)\|_2 \left\|\frac{\partial \bar{x}(0)}{\partial \theta_i}\right\|_2 \end{aligned} \tag{2.17}$$

where P is the Lyapunov matrix and μ depends on $\bar{A}$ as in Lemma 1.

Proof. Using the product rule for the derivative of $\zeta(t, \theta)$ in equation (2.7), we can prove the desired result by combining the results from Theorem 1 and 2 and working out the algebra for other terms that appears in the total derivative equation. □

Nonlinear dynamics — Model reduction robustness

For nonlinear system dynamics, the approach above to derive the bound on the robustness guarantee does not work because we cannot obtain the error dynamics analytically as was possible for linear dynamics in equation (2.7). An alternate approach for deriving a bound on the sensitivity of the error is using local sensitivity analysis. Consider the following nonlinear dynamics of the full system

$$\dot{x} = f(x, \theta), \quad y = Cx, \quad x(0) = x_0. \tag{2.18}$$

The reduced nonlinear model is given using similar notation

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \theta), \quad \hat{y} = \hat{C}\hat{x}, \quad \hat{x}(0) = \hat{x}_0. \tag{2.19}$$

For both the full and the reduced dynamics, we assume that $f : E \subset \mathbb{R}^n \to \mathbb{R}^n$, $\hat{f} : \hat{E} \subset \mathbb{R}^{\hat{n}} \to \mathbb{R}^{\hat{n}}$ are locally Lipschitz functions and initial conditions $x(0) \in E$, $\hat{x}(0) \in \hat{E}$. The local Lipschitz continuity gives us that the solutions $x(t)$ and $\hat{x}(t)$ exist and are unique for a finite time interval. We further assume that equilibrium points $x^* \in \mathbb{R}^n$, $\hat{x}^* \in \mathbb{R}^{\hat{n}}$ exist and there is no finite escape time. See [82, Corollary 2.5] for the sufficient smoothness conditions that are needed on the system dynamics for these assumptions to hold. We make these assumptions throughout this chapter for any nonlinear function describing the system dynamics.

Theorem 3. *For the structured model reduction of the nonlinear dynamical system (2.18) to the reduced system (2.19) by using time-scale separation and quasi-steady-state approximation, the norm of the sensitivity of the error in model reduction, S_ζ is bounded above by*

$$\left\|S_\zeta\right\|_2^2 \leq \lambda_{\max_{\bar{x}}}(P(\bar{x}))\left\|\bar{S}_0\right\|_2^2 + 2\int_0^\infty \left\|\bar{Z}^T P(\bar{x})\bar{S}\right\|_2 dt + \lambda_{\max_{\bar{x}}}(\dot{P}(\bar{x}))\int_0^\infty \left\|\bar{S}\right\|_2^2 dt \tag{2.20}$$

where $P(\bar{x})$ is a matrix that solves the Lyapunov equation $\bar{J}(\bar{x})^T P(\bar{x}) + P(\bar{x})\bar{J}(\bar{x}) = -\bar{C}^T\bar{C}$ at the point $\bar{x}(t) = \bar{x}$ in the augmented nonlinear system trajectory. Here $\bar{J}(\bar{x})$ is the Jacobian matrix at $\bar{x}$, $\bar{Z}$ is the sensitivity to parameter, and $\bar{S}$ is the sensitivity coefficients vector of the augmented system, given by:

$$\bar{J}(\bar{x}) = \begin{bmatrix} \frac{\partial f}{\partial x} & 0 \\ 0 & \frac{\partial \hat{f}}{\partial \hat{x}} \end{bmatrix}, \quad \bar{Z} = \begin{bmatrix} \frac{\partial f}{\partial \theta_i} \\ \frac{\partial \hat{f}}{\partial \theta_i} \end{bmatrix}, \quad \bar{S} = \begin{bmatrix} \frac{\partial x}{\partial \theta_i} \\ \frac{\partial \hat{x}}{\partial \theta_i} \end{bmatrix}.$$

Proof. At the point $\bar{x}(t) = \bar{x}$ in the augmented nonlinear system trajectory, write the sensitivity system equations [80] for a parameter $\theta_i \in \theta$:

$$\dot{\bar{S}} = \bar{J}(\bar{x})\bar{S} + \bar{Z}, \quad S_\zeta = \bar{C}\bar{S}.$$

For the norm of the sensitivity of the error, we can write

$$\|S_\zeta\|_2^2 = \int_0^\infty \bar{S}\bar{C}^T\bar{C}\bar{S}dt.$$

For every $\bar{x}$, given that there exists a matrix $P(\bar{x}) = P^T(\bar{x}) \succ 0$ such that $\bar{J}(\bar{x})^T P(\bar{x}) + P(\bar{x})\bar{J}(\bar{x}) = -\bar{C}^T\bar{C}$, consider a function $V(\bar{S}) = \bar{S}^T P(\bar{x})\bar{S}$. Differentiating this function with respect to time, we have that

$$\frac{dV}{dt} = \bar{S}^T(\bar{J}^T P + P\bar{J})\bar{S} + \bar{S}^T\dot{P}(\bar{x})\bar{S} + (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}).$$

For simplicity of exposition, we denote $P(\bar{x})$ as P noting that this is a state-dependent Lyapunov matrix, similarly $\bar{J}$ is a state-dependent Jacobian matrix of the augmented nonlinear system evaluated at $\bar{x}(t) = \bar{x}$. Integrating the expression above from 0 to ∞ and then substituting the expression for $\|S_\zeta\|_2^2$, we get

$$\|S_\zeta\|_2^2 = \int_0^\infty -\frac{dV}{dt}dt + \int_0^\infty \bar{S}^T\dot{P}\bar{S}dt + \int_0^\infty (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt,$$

$$\|S_\zeta\|_2^2 = -\lim_{t\to\infty} V(\bar{S}(t)) + V(\bar{S}(0)) + \int_0^\infty \bar{S}^T\dot{P}\bar{S}dt + \int_0^\infty (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt.$$

Since, P is a positive semi-definite matrix, V will be a non-negative function [74]. Using this fact and denoting $\bar{S}(0) := \bar{S}_0$, we have the inequality

$$\|S_\zeta\|_2^2 \le \bar{S}_0^T P\bar{S}_0 + \int_0^\infty \bar{S}^T\dot{P}\bar{S}dt + \int_0^\infty (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt. \tag{2.21}$$

For the first part of this equation, we can write

$$\left\|\bar{S}_0^T P\bar{S}_0\right\| \le \lambda_{\max_{\bar{x}}}(P)\left\|\bar{S}_0\right\|^2,$$

where we compute the maximum eigenvalue of P over all points $\bar{x}$. Note that if the initial conditions are independent of all model parameters then $\bar{S}_0 = 0$. Similarly, we get,

$$\left\|\int_0^\infty \bar{S}^T\dot{P}\bar{S}dt\right\| \le \lambda_{\max_{\bar{x}}}(\dot{P})\int_0^\infty \left\|\bar{S}\right\|^2 dt.$$

Evaluating the last part in equation (2.21) proves the theorem, combined with the above results:

$$\int_0^\infty \left(\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}\right) dt \le 2\int_0^\infty \left\|\bar{Z}^T P\bar{S}\right\|_2 dt.$$

□

Remark. For computational purposes, we may modify the bound above as

$$\left\|S_\zeta\right\|_2^2 \leq \lambda_{\max_{\bar{x}}}(P)\left\|\bar{S}(0)\right\|_2^2 + 2N\sup_t\left\|\bar{Z}^T P\bar{S}\right\|_2 + N\lambda_{\max_{\bar{x}}}(\dot{P})\sup_t\left\|\bar{S}\right\|_2^2 \tag{2.22}$$

where $N > 0$ is a time at which the system solution is arbitrarily close to the equilibrium point, that is, $\|\bar{x}(N) - \bar{x}^*\| < \epsilon$ for some $\epsilon > 0$ and $\bar{x}^*$ is the equilibrium point for the augmented nonlinear dynamical system. Since we compute $P(\bar{x})$ at every time step, we can also compute $\dot{P}(\bar{x})$ numerically. Further, a direct computation for S_ζ is also possible but the bounds that we give may be used as interpret-able metrics for system performance analysis.

Equivalence of the two results for linear dynamics

A direct comparison of the results in Corollary 2.1 (robustness estimate for linear dynamics) and Theorem 3 (robustness estimate for nonlinear dynamics) is not evident. But for the special case of linear dynamics, we have the closed-form solutions for $\bar{S}(t)$ and $\bar{Z}(t)$. Using these we can evaluate the bound given in Theorem 3 further.

Claim. The bound on the sensitivity of the error in model reduction when obtained using sensitivity analysis approach (as in equation (2.20)) is same as the bound obtained using direct linear analysis approach (as given in equation (2.17)). In particular, we have that,

$$\begin{aligned} 2\int_0^\infty \left\|\left(\frac{\partial \bar{A}}{\partial \theta_i}\bar{x}\right)^T P\bar{S}\right\| dt \leq{}& \frac{1}{4\,|\mu|^3}\left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\|^2 \left\|\bar{C}^T\bar{C}\right\| \|\bar{x}(0)\|^2 \\ &+ \frac{1}{2\,|\mu|^2}\left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\| \left\|\bar{C}^T\bar{C}\right\| \|\bar{x}(0)\| \left\|\frac{\partial \bar{x}(0)}{\partial \theta_i}\right\|. \end{aligned} \tag{2.23}$$

Note that the first term in equation (2.17), for linear system bound, and equation (2.20), for nonlinear system bound, is the same. The first term corresponds to the parametric uncertainty in the initial conditions, which is the same for both the linear and the nonlinear result. Also, the third term in the nonlinear bound in equation (2.20) is equal to zero for the case of linear system dynamics as the Lyapunov matrix, $P(\bar{x})$ is independent of the state value, that is, $P(\bar{x}) = P$. So, the derivative of P is zero. Hence we have removed the first and the third term in the comparison above. So, we show in the proof below that the sensitivity analysis based method to compute the bound on $\left\|\bar{S}(\zeta)\right\|^2$ (equation (2.20)) also leads to exactly the same bound as given in the direct linear analysis approach in equation (2.17) for the case of linear dynamics.

Proof. To prove the above claim, we start by evaluating the different parts of the left hand side expression using the closed-form solutions for linear dynamics. First, note that from the sensitivity equation [80] for a linear system we have that

$$\dot{\bar{S}} = \bar{A}\bar{S} + \frac{\partial \bar{A}}{\partial \theta_i}\bar{x}(t).$$

Solving the equation above for $\bar{S}(t)$ and taking the norm we get

$$\begin{aligned} \left\|\bar{S}(t)\right\| &\leq \left\|e^{\bar{A}t}\bar{S}(0)\right\| + \left\|\int_0^t e^{\bar{A}(t-\tau)}\frac{\partial \bar{A}}{\partial \theta_i}\bar{x}(\tau)d\tau\right\| \\ &\leq e^{-|\mu|t}\left\|\bar{S}(0)\right\| + \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\| \left\|\bar{x}(0)\right\| te^{-|\mu|t} \end{aligned} \tag{2.24}$$

using Lemma 1 and 2. Similarly, for $\left\|\bar{Z}\right\|$, we have

$$\left\|\bar{Z}\right\| = \left\|\frac{\partial \bar{A}}{\partial \theta_i}\bar{x}\right\| \leq \left\|\frac{\partial \bar{A}}{\partial \theta_i}\right\| \left\|\bar{x}(0)\right\| e^{-|\mu|t}. \tag{2.25}$$

Finally, for the Lyapunov matrix, we know that

$$\|P\| = \left\|\int_0^\infty e^{\bar{A}^T t}\bar{C}^T\bar{C}e^{\bar{A}t}dt\right\|,$$

using the observability Gramian. Using Lemma 1, we have

$$\|P\| \leq \int_0^\infty e^{-|\mu|t}\left\|\bar{C}^T\bar{C}\right\| e^{-|\mu|t}dt.$$

So,

$$\|P\| \leq \left\|\bar{C}^T\bar{C}\right\| \int_0^\infty e^{-2|\mu|t}dt,$$

which gives us that

$$\|P\| \leq \frac{1}{2|\mu|}\left\|\bar{C}^T\bar{C}\right\|. \tag{2.26}$$

Substituting the equations (2.24), (2.25), and (2.26) into the left hand size of equation (2.23), we get the desired result that proves our claim. □

Although the two approaches give equivalent results for linear dynamics, the advantage with the sensitivity analysis approach is that it is a general method that can be used for nonlinear dynamical systems as well.

Controlled nonlinear systems

Next, we consider the structured model reduction of a controlled nonlinear dynamical system. The nonlinear system dynamics are then given by,

$$\begin{aligned} \dot{x} &= f(x,\theta) + g(x,\theta)u, \\ y &= Cx, \quad x(0) = x_0. \end{aligned} \tag{2.27}$$

The reduced nonlinear model is given using similar notation

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x},\theta) + \hat{g}(\hat{x},\theta)u, \\ \hat{y} &= \hat{C}\hat{x}, \quad \hat{x}(0) = \hat{x}_0. \end{aligned} \tag{2.28}$$

Here, we have assumed a scalar input u for simplicity of exposition. The results that follow can be derived for systems with multiple inputs as well but with more complicated algebra.

Theorem 4. *For the structured model reduction of the nonlinear controlled dynamics (2.27) to the reduced dynamics (2.28), the norm of the sensitivity of the error S_ζ is bounded above by,*

$$\begin{aligned} \left\|S_\zeta\right\|_2^2 \le \lambda_{\max_{\bar{x}}}(P)\left\|\bar{S_0}\right\|^2 + 2\int_0^\infty \left(\left\|\bar{Z}_f^T P\bar{S}\right\| + \left\|u^T\bar{S}^T\bar{J}_g^T P\bar{S}\right\| + \left\|\bar{Z}_g^T P\bar{S}u\right\|\right) dt \\ + \lambda_{\max_{\bar{x}}}(\dot{P})\int_0^\infty \left\|\bar{S}\right\|^2 dt \end{aligned}$$

if there exists $P(\bar{x}) = P(\bar{x})^T \succ 0$ such that $P(\bar{x})\bar{J}_f(\bar{x}) + \bar{J}_f(\bar{x})^T P(\bar{x}) = -\bar{C}^T\bar{C}$ at the point $\bar{x}(t) = \bar{x}$ in the augmented nonlinear system trajectory. Here $\bar{J}_f(\bar{x}), \bar{J}_g(\bar{x})$ and $\bar{Z}_f, \bar{Z}_g$ are the Jacobian and parameter sensitivity matrices of augmented nonlinear functions $\bar{f}$ and $\bar{g}$, respectively.

Proof. The proof follows similar to the proof of the previous theorem by defining a function $V(\bar{S}) = \bar{S}^T P(\bar{x})\bar{S}$ at every point $\bar{x}$ in the system trajectory and calculating the bound for $\left\|S_\zeta\right\|$.

For the augmented system we can write the sensitivity coefficients as

$$\bar{S} = \frac{\partial \bar{x}}{\partial \theta_i},$$

where $\theta_i \in \theta$. Using chain rule at point $\bar{x}(t) = \bar{x}$ in the system trajectory, we can derive the sensitivity system equation given by

$$\dot{\bar{S}} = \left(\bar{J}_f(\bar{x})\bar{S} + \bar{Z}_f\right) + \left(\bar{J}_g(\bar{x})\bar{S} + \bar{Z}_g\right)u,$$

for a scalar u and Jacobian matrices J_f and J_g are state-dependent. Consider a function $V(\bar{S}) = \bar{S}^T P(\bar{x})\bar{S}$ for $P(\bar{x}) = P(\bar{x})^T > 0$ that satisfies the conditions given in the theorem statement. Note that we drop the $\bar{x}$ notation from $\bar{J}_f, \bar{J}_g, P$ for simplicity. Taking the derivative of V with respect to time, we can write,

$$\frac{dV}{dt} = \dot{\bar{S}}^T P\bar{S} + \bar{S}^T P\dot{\bar{S}} + \bar{S}^T \dot{P}\bar{S}.$$

Substituting for $\dot{\bar{S}}$ we get

$$\begin{aligned}\frac{dV}{dt} = \bar{S}^T\left(P\bar{J}_f + \bar{J}_f^T P\right)\bar{S} + u^T\bar{S}^T\bar{J}_g^T P\bar{S} + \bar{S}^T P\bar{J}_g\bar{S}u + \left(\bar{S}^T P\bar{Z}_f + \bar{Z}_f^T P\bar{S}\right)\\ + \left(\bar{S}^T P\bar{Z}_g u + u^T\bar{Z}_g^T P\bar{S}\right) + \bar{S}^T\dot{P}\bar{S},\end{aligned}$$

Now if there exists a matrix $P = P^T$ such that

$$P\bar{J}_f + \bar{J}_f^T P = -\bar{C}^T\bar{C},$$

then we get the following bound by manipulating the 2-norm of S_ζ and denoting $\bar{S}(0) := \bar{S}_0$,

$$\left\|S_\zeta\right\|_2^2 \leq \bar{S}_0^T P\bar{S}_0 + 2\int_0^\infty \left(\left\|u^T\bar{S}^T\bar{J}_g^T P\bar{S}\right\| + \left\|\bar{Z}_f^T P\bar{S}\right\| + \left\|u^T\bar{Z}_g^T P\bar{S}\right\|\right)dt + \int_0^\infty \bar{S}^T\dot{P}\bar{S}dt.$$

We can simplify the above to get the desired result,

$$\begin{aligned}\left\|S_\zeta\right\|_2^2 \leq \lambda_{\max_{\bar{x}}}(P)\left\|\bar{S}_0\right\|^2 + 2\int_0^\infty \left(\left\|u^T\bar{S}^T\bar{J}_g^T P\bar{S}\right\| + \left\|\bar{Z}_f^T P\bar{S}\right\| + \left\|u^T\bar{Z}_g^T P\bar{S}\right\|\right)dt\\ + \lambda_{\max_{\bar{x}}}(\dot{P})\int_0^\infty \left\|\bar{S}\right\|^2 dt.\end{aligned}$$

□

The results above assumed that the outputs of the system are linearly related to the states $y = Cx$, however, we can derive similar results even without this assumption.

Theorem 5. *For the system dynamics in equation (2.27) with output dynamics given by $y = h(x, \theta)$ and the reduced model dynamics given in equation (2.28) with output dynamics given by $\hat{y} = \hat{h}(\hat{x}, \theta)$, the norm of the sensitivity of error is bounded above by,*

$$\begin{aligned}\left\|S_\zeta\right\|_2^2 \leq \lambda_{\max_{\bar{x}}}(P)\left\|\bar{S}(0)\right\|^2 + 2\int_0^\infty \left(\left\|\bar{Z}_f^T P\bar{S}\right\|_2 + \left\|u^T\bar{S}^T\bar{J}_g^T P\bar{S}\right\| + \left\|\bar{Z}_g^T P\bar{S}u\right\|\right)dt\\ + 2\left\|\bar{Z}_h\right\|_2^2 + \lambda_{\max_{\bar{x}}}(\dot{P})\int_0^\infty \left\|\bar{S}\right\|^2 dt\end{aligned}$$

if there exists $P(\bar{x}) = P(\bar{x})^T \succ 0$ *such that* $\bar{J}_f(\bar{x})^T P(\bar{x}) + P(\bar{x})\bar{J}_f(\bar{x}) = -\bar{C}_1^T\bar{C}_1$, *where*

$$\bar{C}_1 = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} J_h(\bar{x}) & 0 \\ 0 & \hat{J}_h(\bar{x}) \end{bmatrix} \triangleq C_e\bar{J}_h(\bar{x}),$$

and $\bar{J}_*(\bar{x}), \bar{Z}_*$ *refer to the Jacobian and the parameter sensitivity matrices for each nonlinear function, respectively, at a point* $\bar{x}(t) = \bar{x}$ *in the system trajectory given by:*

$$\bar{J}_*(\bar{x}) = \begin{bmatrix} J_*(x) & 0 \\ 0 & \hat{J}_*(\hat{x}) \end{bmatrix}, \quad \bar{Z}_* = \begin{bmatrix} Z_* \\ \hat{Z}_* \end{bmatrix}, \quad * : f, g, h.$$

Proof. For every point $\bar{x}(t) = \bar{x}$, we write the sensitivity of the error in model reduction for a parameter $\theta_i \in \theta$ using chain rule:

$$S_\zeta = \frac{d(h - \hat{h})}{d\theta_i} = \left(\frac{\partial h}{\partial x}\right)\left(\frac{\partial x}{\partial \theta_i}\right) + \left(\frac{\partial h}{\partial \theta_i}\right) - \left[\left(\frac{\partial \hat{h}}{\partial \hat{x}}\right)\left(\frac{\partial \hat{x}}{\partial \theta_i}\right) + \left(\frac{\partial \hat{h}}{\partial \theta_i}\right)\right].$$

Define the Jacobians $J_h(\bar{x}), \hat{J}_h(\bar{x})$, parameter sensitivity matrices $Z_h, \hat{Z}_h$ for h and $\hat{h}$, respectively, and substitute back to write,

$$\begin{aligned} S_\zeta &= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} J_h(x) & 0 \\ 0 & \hat{J}_h(\hat{x}) \end{bmatrix} \begin{bmatrix} S \\ \hat{S} \end{bmatrix} + \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} Z_h \\ \hat{Z}_h \end{bmatrix}, \\ S_\zeta &= \bar{C}_1\bar{S} + C_e\bar{Z}_h, \end{aligned}$$

where

$$\bar{C}_1 := \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} J_h(x) & 0 \\ 0 & \hat{J}_h(\hat{x}) \end{bmatrix}, \quad C_e := \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

Now, consider a function $V(\bar{S}) = \bar{S}^T P(\bar{x})\bar{S}$ and proceed in a similar way as in the proof of previous results to write $\|S_\zeta\|_2^2$,

$$\begin{aligned} \|S_\zeta\|_2^2 &= \int_0^\infty \bar{S}^T\bar{C}_1^T\bar{C}_1\bar{S}dt + \int_0^\infty \bar{Z}_h^T C_e^T C_e\bar{Z}_h dt \\ &= \int_0^\infty \bar{S}^T\bar{C}_1^T\bar{C}_1\bar{S}dt + 2\left\|\bar{Z}_h\right\|_2^2. \end{aligned}$$

Using the result from Theorem 4 and $\bar{J}_f(\bar{x})^T P(\bar{x}) + P(\bar{x})\bar{J}_f(\bar{x}) = -\bar{C}_1^T\bar{C}_1$, we get the desired result. □

Remark. For all results on robustness estimate, it is possible to further simplify the bounds using the method in [66] to express this bound only in terms of $\hat{x}$, the reduced state variables and x_c, the collapsed state variables. This might be useful to generate algorithms to efficiently compute the robustness bounds.

Input-output mapping

For forced nonlinear systems, in addition to the error and the robustness metric, it is also important to consider the input-output mapping so that the response of a reduced model for an input is similar to that of the full model. If the mapping from $u \mapsto y$ is linear, the computation of induced system norms is a well studied topic, see for example [83]. However, for general nonlinear systems, this is still an active research area [84] with results only available under certain structural conditions on the system dynamics. Similar to the gap-metric for linear systems [85], [86], there have been a few results on computation of the gap metric for nonlinear systems [87]. We can use similar computational results in our model reduction procedure by assigning the gap metric to each nonlinear reduced-order model.

On the other hand, if we are only interested in the response to an input at steady-state or at a fixed number of points in the response, then we can linearize the dynamics at these points and assess the induced system norm for each reduced model and use this as a metric while choosing a reduced-order model. For the system operator $H : u \mapsto y$, we define,

$$\gamma(H) := \sup_{u \neq 0} \frac{\|Hu\|}{\|u\|}. \tag{2.29}$$

2.4 A new Python package for automated model reduction — AutoReduce

Given a well-behaved full system model (linear or nonlinear) under the stability assumptions, we develop an automated model reduction pipeline based on QSSA and our results on the robustness of model reduction. In the first step, we solve the time-scale separation problem to obtain all possible reduced models. Then, we may numerically bound the error in the model reduction for each reduced model as shown in [66] or compute an error metric ($\|\zeta\| = \|y - \hat{y}\|$) by directly simulating the system. The latter method also works for nonlinear dynamical systems. We may reject any reduced model at this stage that does not exhibit the desired level of error performance. However, to make a clear choice of a reduced model, we compute a robustness metric for each reduced model using the results in the previous section. Finally, if the system is controlled then we may use the input-output mapping metric as well for the linearized system dynamics to determine the performance of a reduced model in terms of the input-output response. Note that it is not required to explore the space of all possible reduced models to compute any of the proposed metrics. This is especially important for large system models where symbolically computing all possible reduced models would be computationally infeasible. Hence, for large system models, we can compute a heuristically chosen set of reduced models. Then, we can compute the error, robustness, and input-output mapping metrics as desired for the reduced models of interest.

We provide an implementation of these model reduction tools in a package called `AutoReduce`. This package is available as an open-source project on GitHub [88]. The software is based on Python Sympy [89] and works by loading a Systems Biology Markup Language (SBML) [90] model to import any biological system model. The following important tools are available in this software:

Conservation laws

Using conservation laws, we can eliminate states that are conserved from the imported SBML model. We integrate this approach in our method similar to the method in [91]. The conservation laws can be explicitly defined or automatically computed in `AutoReduce`. These are used in the software package accordingly to eliminate state variables and compute the reduced-order models.

Time-scale separation

To solve the time-scale separation problem for a given model, the package methods can be used to set the dynamics of the given states to collapse (x_c) to zero and to automatically substitute back into the dynamics for the reduced state variables ($\hat{x}$). This method automates the QSSA procedure and can be used to compute QSSA based reduced models without specifying parameter values.

Comparison metrics

The performance metrics discussed above are implemented in this package as well. For any pair of full model and reduced model, metrics such as the norm of the error ($||\zeta||$) or the robustness metric ($\left\|S_\zeta\right\|$) can be computed.

2.5 Biomolecular system examples

In this section, we apply our model reduction robustness estimate results to biomolecular systems to demonstrate the utility of our approach.

Enzymatic reaction dynamics

For the enzymatic reaction system, we can write a chemical reaction network model using mass-action kinetics for the following reactions:

$$E + S \underset{d}{\overset{a}{\rightleftharpoons}} C \xrightarrow{k} P + E. \tag{2.30}$$

The full mass-action kinetics based model with four species is given by:

$$\frac{dS}{dt} = -aES + dC, \quad \frac{dC}{dt} = aES - (d+k)C$$
$$\frac{dE}{dt} = -aES + dC + kC, \quad \frac{dP}{dt} = kC.$$

For this example, we discuss the standard model reduction approach using the singular perturbation method along with our automated model reduction approach. Recall that for singular perturbation theory approach it is important to separate the dynamics analytically according to the time scales, which might not be possible for general system dynamics. But for this example, it is possible to derive the reduced model using singular perturbation theory as well.

We write the conservation laws for this system as $E = E_{\text{tot}} - C$ and $S + C + P = S_{\text{tot}}$, where E_{tot} and S_{tot} are the total enzyme and substrate concentrations, respectively. We also specify that the output of interest for this system is concentration of P, the product species.

For singular perturbation approach, we manually simplify [40, Ch.3] the system dynamics to write the system with conservation laws in the required form (2.3):

$$\epsilon \frac{dC}{dt} = \frac{k}{K_d} \left(E_{\text{tot}} - C\right) \left(S_{\text{tot}} - C - P\right) - kC - \epsilon kC,$$

$$\frac{dP}{dt} = kC,$$

where $K_d = d/a$ and the small parameter $\epsilon := k/d$. When $\epsilon \to 0$, we obtain the reduced model. For singular perturbation theory based model reduction, a singular perturbation margin (SPM) has been proposed in the literature to assess the robustness of model reduction [92]. The SPM evaluates the maximum value of ϵ such that the singularly perturbed system dynamics are unstable. But for biologically relevant parameters the dynamics given above do not become unstable and hence we do not have a metric to compute the robustness of model reduction directly.

The automated model reduction package first eliminates the two state-variables E and S based on the conservation laws. In the next step, it solves for time-scale separation to obtain a possible reduced-order model. For this reduced model, we computed the error metric ($\|\zeta\| = 4.2 \times 10^{-3}$) as well as the robustness metric over the model initial conditions and parameters [88]. If the system performance is satisfactory, then we can conclude that the final reduced model is given by:

$$\frac{dP}{dt} = K_L \bar{C},$$

where $K_L = kE_{\text{tot}}$ is the lumped parameter and $\bar{C}$ is a function of E_{tot}, S_{tot} and model parameters given by

$$\bar{C} = \frac{S_{\text{tot}} - P}{S_{\text{tot}} + K_m}$$

where $K_m = (d + k)/a$. All of the simulations, and code required for computations for this example is available at [88]. An important distinction with our automated computational method to derive reduced models is that any spurious conditions are easier to catch. For this example, it is shown in [93, Ch.3] that the commonly used model derived above fails to capture the true dynamics under certain parameter regimes. The robustness properties of the model reduction would inform the analysis whenever such a condition may occur.

For all biologically relevant parameter values in the enzymatic reaction system, the response does not become unstable. So, we may not be able to use the singular perturbation margin (SPM) to estimate robustness of this model reduction [92]. However, our method is still applicable as it depends on computing the sensitivity of the system states to the parameters.

Gene expression — Design space exploration

With this example, we demonstrate that a two-state ribosome and protein model can robustly capture the chemical reaction dynamics of gene expression. We analytically derive phenomenological models for gene expression with exactly known mappings to the mechanistic details. We explore the modeling assumptions of time-scale separation [94]–[96], conservation laws [91], [93], [97] and prove the robustness of various models under certain parametric conditions.

For gene expression, a two-state model is commonly used [10], [11], [98]–[100] in the literature that models the dynamics of the mRNA (T) and the protein concentration (X) as a function of the DNA copy number (G) and regulatory effects:

$$\frac{dT}{dt} = k_{tx} f_{tx}(G,\ \cdot) - d_T T$$
$$\frac{dX}{dt} = k_{tl} f_{tl}(T) - d_X X,$$

where k_{tx} is defined as the transcription rate and k_{tl} is the translation rate. Similarly, d_T and d_X are the degradation and dilution parameters for the transcript and the protein, respectively. The function $f_{tx}(\cdot)$ is usually a Hill function dependent on the mechanism of transcriptional regulation (activation or repression). For constitutive expression, this is assumed to be a constant function of the DNA copy number, $f_{tx}(G) = kG$. Similarly, $f_{tl}(\cdot)$ could be a constant or a Hill function dependent on the transcriptional regulation mechanism [93]. Clearly, the parameters in this model and any parameters in the Hill functions all have empirical meanings but an analytical relationship with the mechanistic reaction rates is usually obscured [33]. Moreover, a closer analysis would show that such phenomenological models are only valid under certain assumptions and parameter regimes.

The full CRN model

The full chemical reaction network (CRN) model for the expression of protein X from a single gene G is described in Table 2.1. In this CRN, the gene G is transcribed by RNA polymerase (P) to an mRNA transcript T via a complex (C_1) formation reaction. Then, the transcript T binds to the ribosome R to form the second complex C_2, which then translates to express the protein X. Under the assumption of mass-action kinetics for all reactions, the ordinary differential equation (ODE) model can be derived as shown on the right in Table 2.1. We refer to this as the full CRN model for the rest of this chapter.

Reduced-order modeling

For the CRN model, the first step is to solve for the conserved quantities in the model. We assume that the total RNA polymerase in the system and the total ribosomes remain conserved. From Table 2.1, we observe that

$$\frac{dP}{dt} + \frac{dC_1}{dt} = 0, \quad \frac{dR}{dt} + \frac{dC_2}{dt} = 0.$$

Hence, for constants P_{tot} and R_{tot}, we have that $P_{\text{tot}} = P + C_1$ and $R_{\text{tot}} = R + C_2$. These conservation laws can be used to eliminate C_1 and C_2. `AutoReduce` can

Table 2.1: Gene expression model.

CRN	ODE model	Nominal Parameter Values
$G + P \underset{k_{up}}{\overset{k_{bp}}{\rightleftharpoons}} C_1$	$\frac{dP}{dt} = \left(k_{up} + k_{tx}\right) C_1$	$k_{bp} = 80,$
	$-k_{bp}GP$	$k_{up} = 2$
$C_1 \xrightarrow{k_{tx}} G + P + T$	$\frac{dC_1}{dt} = k_{bp}GP - \left(k_{up} + k_{tx}\right) C_1$	$k_{tx} = 0.5$
$T + R \underset{k_{ur}}{\overset{k_{br}}{\rightleftharpoons}} C_2$	$\frac{dT}{dt} = k_{tx}C_1 + (k_{ur} + k_{tl})\, C_2$	$k_{br} = 80,$
	$-k_{br}TR - d_T T$	$k_{ur} = 2$
$C_2 \xrightarrow{k_{tl}} T + R + X$	$\frac{dR}{dt} = (k_{ur} + k_{tl})\, C_2 - k_{br}TR$	$k_{tl} = 0.5$
$T \xrightarrow{d_T} \varnothing$	$\frac{dC_2}{dt} = k_{br}TR - (k_{ur} + k_{tl})\, C_2$	$d_T = 0.5$
$X \xrightarrow{d_X} \varnothing$	$\frac{dX}{dt} = k_{tl}C_2 - d_X X$	$d_X = 0.01$

obtain these conservation laws automatically by using symbolic computations in the differential equations. After substituting the conservation laws using `AutoReduce`, we obtain the following reduced-order model,

$$\begin{aligned}
\frac{dP}{dt} &= (k_{tx} + k_{up})(P_{\text{tot}} - P) - k_{bp}GP,\\
\frac{dT}{dt} &= k_{tx}(P_{\text{tot}} - P) + (k_{tl} + k_{ur})(R_{\text{tot}} - R) - k_{br}RT - d_T T,\\
\frac{dR}{dt} &= (k_{tl} + k_{ur})(R_{\text{tot}} - R) - k_{br}RT,\\
\frac{dX}{dt} &= k_{tl}(R_{\text{tot}} - R) - d_X X.
\end{aligned} \tag{2.31}$$

Next, we obtain various reduced models under different time-scale separation assumptions. Recall that we denote all reduced-order model variables with a hat to differentiate the corresponding variable in the full model, for example, in a reduced model the protein species will be represented as $\hat{X}$ and the corresponding species in the full model is denoted by X. All variables denote the concentrations for each species and parameters take appropriate units. Furthermore, we define the following lumped parameter notations which appear as Hill function activation parameters in the reduced model expressions that we derive next.

$$K_0 := \frac{k_{tl} + k_{ur}}{k_{br}}, \qquad K_1 := \frac{k_{tx} + k_{up}}{k_{bp}}, \qquad K_d := \frac{k_{tl} + k_{ur}}{d_T} \tag{2.32}$$

The mRNA transcript ($\hat{T}$) and protein ($\hat{X}$) model

Under the assumption that the free ribosomes and the RNA polymerase dynamics are at QSS (that is $\dot{R} = 0, \dot{P} = 0$), we obtain the following model with only the mRNA transcript and the protein dynamics as a function of the DNA copy number G:

$$\begin{aligned} \frac{d\hat{T}}{dt} &= k_{tx} P_{\text{tot}} \left(\frac{G}{K_1 + G} \right) - d_T \hat{T} \\ \frac{d\hat{X}}{dt} &= k_{tl} R_{\text{tot}} \frac{\hat{T}}{K_0 + \hat{T}} - d_X \hat{X}. \end{aligned} \tag{2.33}$$

The free ribosome ($\hat{R}$) and protein ($\hat{X}$) model

Under the assumption that the mRNA transcript and the RNA polymerase dynamics are at QSS (that is $\dot{T} = 0, \dot{P} = 0$), we obtain the following model with only the free ribosome and the protein dynamics:

$$\frac{d\hat{R}}{dt} = \frac{d_T (R_{tot} - \hat{R})}{K_0^{-1} \hat{R} + K_d^{-1}} - k_{tx} P_{tot} \left(\frac{G}{K_1 + G} \right) \left(\frac{\hat{R}}{\hat{R} + \frac{K_0}{K_d}} \right)$$

$$\frac{d\hat{X}}{dt} = k_{tl} \left(R_{\text{tot}} - R \right) - d_X \hat{X}. \tag{2.34}$$

Similar to the ribosome-protein ($[\hat{R}, \hat{X}]$) and the mRNA transcript-protein models ($[\hat{T}, \hat{X}]$), it is possible to derive the polymerase-protein ($[\hat{P}, \hat{X}]$) and the only protein model ($[\hat{X}]$). The detailed equations for these two models are not presented here for brevity, but their performance is shown in Figures 2.2, 2.3, 2.5, and 2.3.

The available free ribosome ($\hat{R}_\Delta$) and protein ($\hat{X}$) model

We define the available free ribosomes in the gene expression system as $\hat{R}_\Delta = R_{\text{tot}} - \hat{R}$. Substituting $\hat{R}$ for $R_{\text{tot}} - \hat{R}_\Delta$, we can derive a new reduced-order model from the $\begin{bmatrix} \hat{R} & \hat{X} \end{bmatrix}$ model — the $\hat{R}_\Delta$ model:

$$\frac{d\hat{R}_\Delta}{dt} = k_{tx} P_{tot} \left(\frac{G}{K_1 + G} \right) \left(\frac{R_{\text{tot}} - \hat{R}_\Delta}{R_{\text{tot}} - \hat{R}_\Delta + \frac{K_0}{K_d}} \right) - \left(\frac{d_T}{K_0^{-1} \left(R_{\text{tot}} - \hat{R}_\Delta \right) + K_d^{-1}} \right) \hat{R}_\Delta,$$

$$\frac{d\hat{X}}{dt} = k_{tl} \hat{R}_\Delta - d_X \hat{X}. \tag{2.35}$$

This model simplifies the robustness analysis that follows. Note that the $\hat{R}_\Delta$ model closely resembles the commonly used gene expression model and all of its terms are

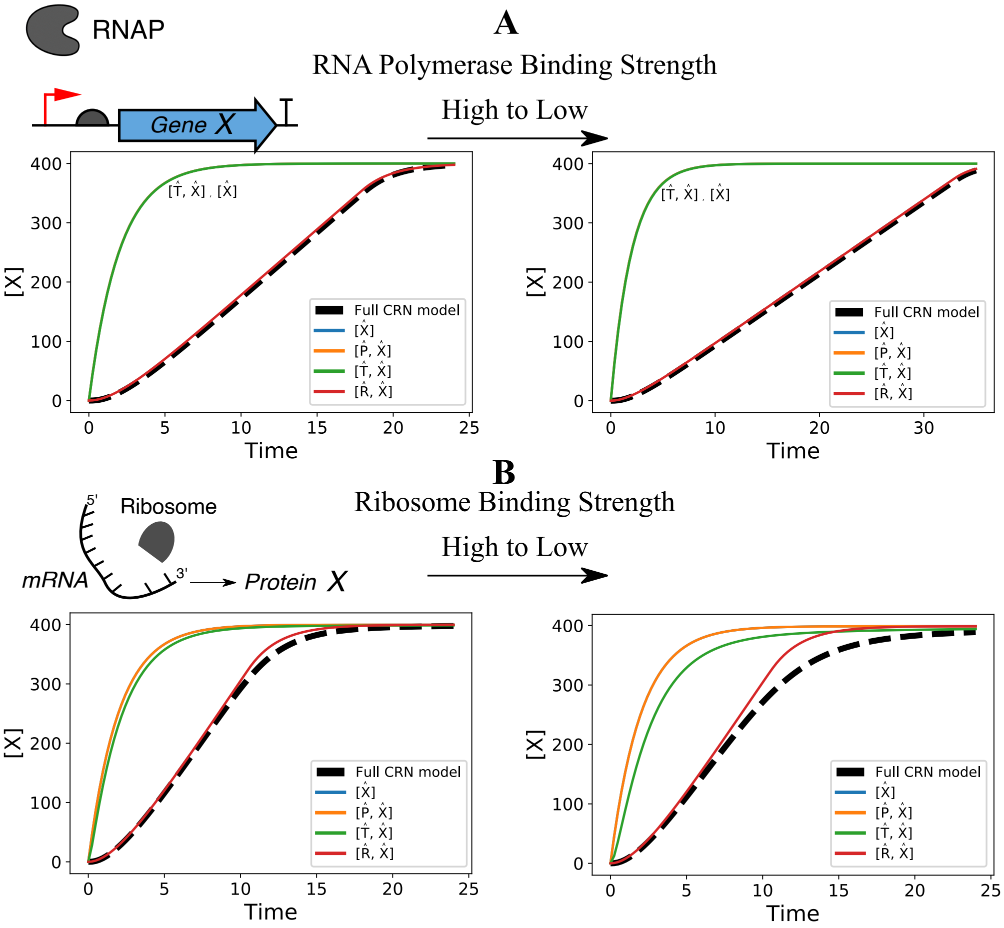


Figure 2.2: Performance of the gene expression models under different biological conditions. **(A)** We observe that for weaker binding of RNA polymerase to the promoter region of the DNA, the time-response of the full CRN model is slower as it takes a longer time to reach steady-state. The mathematical model with only the mRNA transcript and protein dynamics is unable to capture this effect since this binding reaction is assumed to be at quasi-steady-state in this model. On the other hand, the $\hat{R}_\Delta$ model, which describes the dynamics of the free ribosome and the protein is able to capture the effect. Note that the performance of the $[\hat{R}, \hat{X}]$ and the $\hat{R}_\Delta$ model are equivalent. **(B)** With decreasing ribosome binding strength, none of the reduced models perfectly capture the full CRN dynamics, but still, the ribosome-protein model is the closest in error performance to the full model.

exactly similar to the mRNA transcript and protein model but scaled by a ribosome count factor. We give three statements regarding the error performance analysis and the tightness of the robustness guarantees for the reduced models that we derived above.

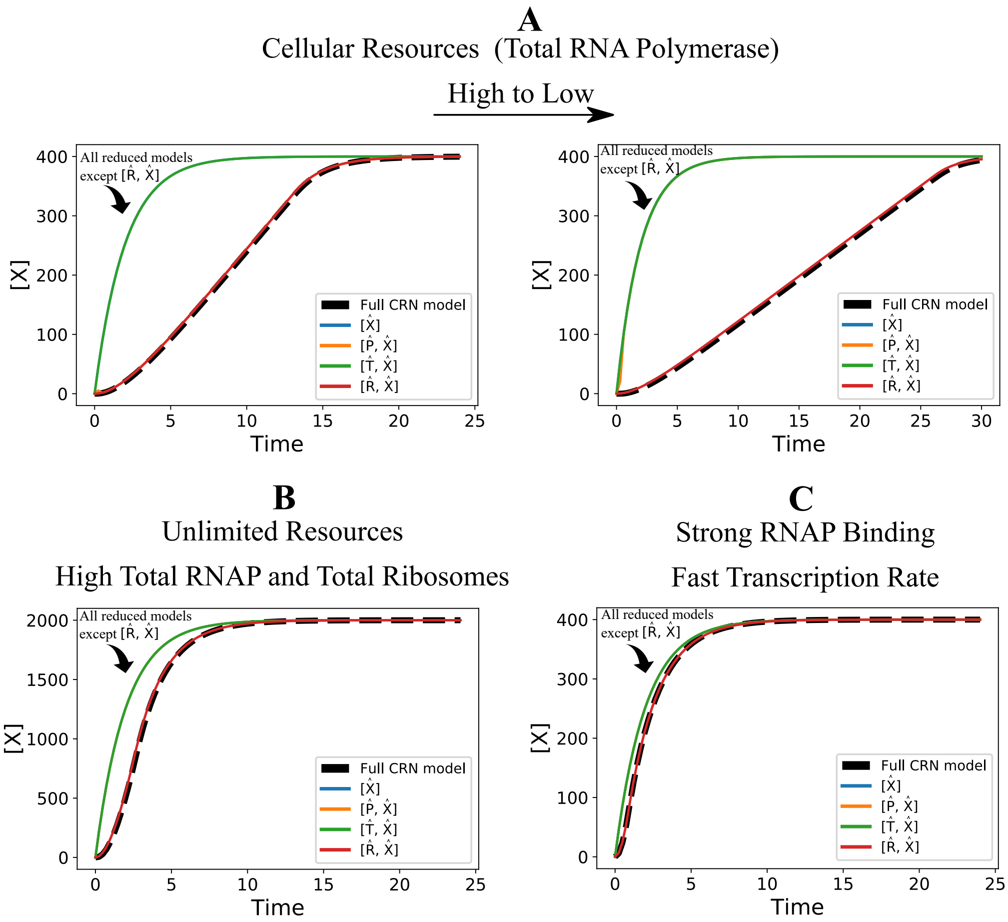


Figure 2.3: Performance of the gene expression models under different biological conditions. **(A)** With decreasing total RNA polymerase, the time-response of the full model is slower and only the ribosome-protein ($[\hat{R}, \hat{X}]$) model is able to account for this effect. **(B)** With a very high amount of total RNA polymerase and the total ribosome count in the system, all models reach steady-state faster. **(C)** Under strong RNA polymerase binding to the DNA and high transcription rate, we see that all models exhibit good error performance. These can be understood as the ideal conditions under which using a one-state protein dynamics model is also justified.

Gene expression – Robustness statement 1

For the gene expression CRN model in Table 2.1, it is not possible to derive reduced models using singular perturbation theory since it is not clear how we could transform the system dynamics to the standard form (as in equation (2.3)). So, we used QSSA to derive accurate reduced models in certain parameter regimes as shown in [101]. In particular, we consider the reduced model with the mRNA and protein dynamics to justify the following statement.

Statement 1. *For a nonlinear system, the linear analysis approach in Theorem 2 gives a tighter robustness guarantee than the direct nonlinear approach in Theo-*

rem 3.

Justification. For the reduced model in equation (2.33), we have the states $\hat{T}$ and $\hat{X}$ that represent the mRNA and protein, respectively. We can obtain a linearized dynamics at every point in the system trajectory. So for a point $x^* = [P^*, T^*, R^*, X^*]$, we have the following system matrices:

$$A = \left[\begin{array}{cccc} \begin{array}{c} -Gk_{bp} \\ -k_{tx} - k_{up} \end{array} & 0 & 0 & 0 \\ \hline -k_{tx} & \begin{array}{c} -k_{br}R^* \\ -d_T \end{array} & \begin{array}{c} -k_{br}T^* \\ -k_{tl} - k_{ur} \end{array} & 0 \\ \hline 0 & -k_{br}R^* & \begin{array}{c} -k_{br}T^* \\ -k_{tl} - k_{ur} \end{array} & 0 \\ \hline 0 & 0 & -k_{tl} & -d_X \end{array}\right],$$

$$\hat{A} = \begin{bmatrix} -d_T & 0 \\ \frac{R_{\text{tot}} k_{tl}}{T^* + \frac{k_{tl} + k_{ur}}{k_{br}}} & -d_X \end{bmatrix}.$$

We can then derive $\frac{\partial A}{\partial \theta_i}$ for each parameter $\theta_i \in \theta$ to compute the robustness estimate bounds for this system using the result in Theorem 2.

The robustness bound on $\|S_\zeta\|$ for each parameter with time is shown in Figure 2.4. As discussed earlier, an alternative approach is to use the result in Theorem 3 to derive bounds directly for the nonlinear systems. With this method, we get a time-integrated result for $\|S_\zeta\|$, which is of the order of 10^4 whereas we see in Figure 2.4 that the bound on $\|S_\zeta\|$ with the linearization approach is of the order of 10^2. Hence, the linear analysis approach (from Theorem 2) gives less conservative results than the direct nonlinear approach (from Theorem 3) but requires the linearization approximation at every point in the system trajectory, so it may not always be accurate and feasible. □

Gene expression — Robustness statement 2

We discuss the robustness performance of the different reduced models derived above in the following statements.

Statement 2. *The error between the $\hat{R}_\Delta$ model and the full CRN model is robust to perturbations in the binding/unbinding of ribosome to the mRNA transcript*

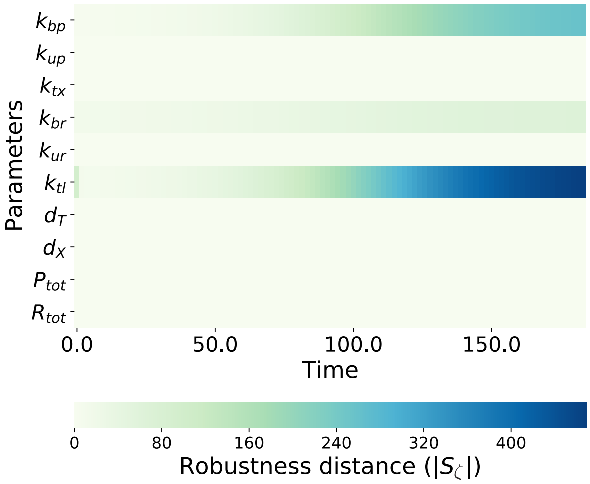


Figure 2.4: Robustness estimates using the result given in Theorem 2 for a gene expression system. Observe that the model reduction error ζ is most sensitive to the translation parameter k_{tl}. This observation is consistent with the result obtained using the nonlinear approach [101, Fig.2].

(k_{br}, k_{ur}), and also total resources (both RNA polymerase, P_{tot} and ribosome, R_{tot}). Only the perturbations in the translation rate (k_{tl}) and the protein degradation (d_X) parameters significantly affect the error performance of the $\hat{R}_\Delta$ model.

Justification. To demonstrate robustness of the $\hat{R}_\Delta$ model, we can look at the sensitivity of the error between the $\hat{R}_\Delta$ model and the full CRN model to various parameter perturbations. Define the error as $\zeta := X - \hat{X}$, where X and $\hat{X}$ represent the protein concentration in the full CRN model and the $\hat{R}_\Delta$ model, respectively. In the case where the error ζ between the two models is within acceptable bounds, we additionally desire that this error is not sensitive to any of the parameters. The sensitivity of the error to perturbation in a parameter θ_i is given by $S_\zeta = \partial\zeta/\partial\theta_i$. Hence, this "fragility metric" S_ζ must be minimized to achieve higher robustness. Analyzing the rate of change of S_ζ with time for different parameters can justify the statement.

To analytically derive $\dot{S}_\zeta$, we use the equation $S_\zeta = \bar{C}\bar{S}$ to write

$$\dot{S}_\zeta = \left(\begin{bmatrix} C & -\hat{C} \end{bmatrix}\right)\left(\begin{bmatrix} J & 0 \\ 0 & \hat{J} \end{bmatrix}\begin{bmatrix} S \\ \hat{S} \end{bmatrix} + \begin{bmatrix} Z \\ \hat{Z} \end{bmatrix}\right).$$

For the model parameters $\theta_i \in \{k_{tx}, k_{bp}, k_{up}, k_{br}, k_{ur}, P_{\text{tot}}, R_{\text{tot}}\}$, we have the dy-

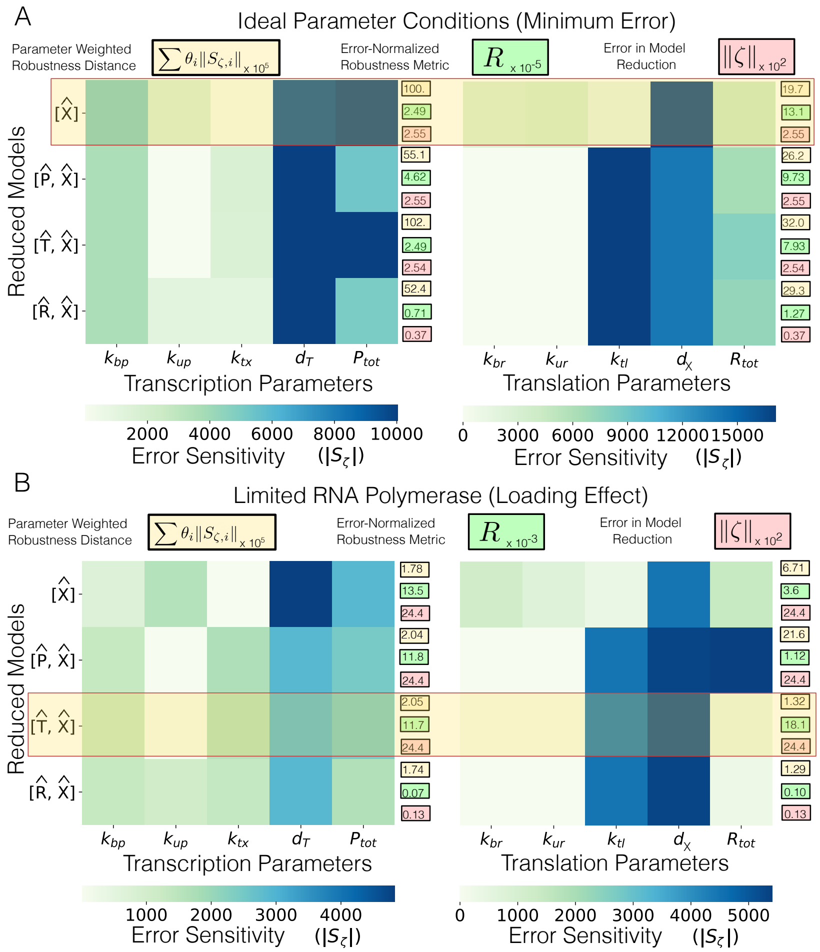


Figure 2.5: Robustness analysis of gene expression models. The figure shows the fragility of different models for all model parameters as computed by the sensitivity of the error in protein X concentration between the full model and the reduced models under different conditions. The weighted robustness distance, the normalized robustness metric, and the normed error in model reduction are labeled for each reduced model in each condition. The combined decision metric r, with weights $w_1 = 1, w_2 = 0.3$, is used to choose one final model in each of the four conditions as shown by the red box. **(A)** Ideal parameter conditions under which all reduced models have similar error performance that is close to the full model. The one-state model with only the protein dynamics has the best performance for the combined metric on error and robustness. **(B)** Under the condition of limited RNA polymerase resources, we get that the mRNA transcript-protein model has the best performance.

namics of S_ζ given by

$$\dot{S}_\zeta = k_{tl} \frac{\partial \left(\hat{R}_\Delta - \hat{R}_\Delta\right)}{\partial \theta_i} - d_X S_\zeta. \tag{2.36}$$

For the protein degradation rate $\theta_i = d_X$, we have

$$\dot{S}_\zeta = k_{tl} \frac{\partial \left(\hat{R}_\Delta - \hat{R}_\Delta\right)}{\partial d_X} - \zeta - d_X S_\zeta. \tag{2.37}$$

For the translation rate $\theta_i = k_{tl}$, we have

$$\dot{S}_\zeta = k_{tl} \frac{\partial \left(\hat{R}_\Delta - \hat{R}_\Delta\right)}{\partial k_{tl}} + \left(\hat{R}_\Delta - \hat{R}_\Delta\right) - d_X S_\zeta. \tag{2.38}$$

From equations (2.36) – (2.38), we have that for all positive parameter values and stable models, the dynamics of S_ζ are convergent to a fixed point. More importantly, the fragility metric, S_ζ directly depends on the translation rate k_{tl} and the protein degradation rate d_X, proving the assertion of the statement above. Also, for k_{tl} and d_X, we see that an extra term appears in the error sensitivity dynamics implying higher fragility of the $\hat{R}_\Delta$ model under perturbations to the translation rate and the protein degradation rate. □

Figures 2.2 and 2.3 shows the error performance of the $\hat{R}_\Delta$ model under various biologically plausible parameter conditions and assumptions. The Euclidean norm of S_ζ is plotted in Figures 2.5 and 2.6 to compare the robustness of the $\hat{R}_\Delta$ model alongside other models in different biologically plausible parameter conditions. Using $\left\|S_\zeta\right\|$ we can obtain a weighted sum to compute the robustness distance d_R for each reduced model using equation (2.12). A normalized robustness metric R is also shown for the reduced models. To choose a particular reduced model for each parameter condition, we use a linear combination of the error and the robustness distance defined as:

$$r := w_1 \left\|\zeta\right\| + w_2 d_R, \tag{2.39}$$

where w_1 and w_2 are free parameters that can be chosen to weigh the error and the robustness performance, respectively. This metric can be used as a single scalar to compare the reduced models, lower r implies better performance. Similar to the result above, we give the following statement for the transcript and the protein model.

Statement 3. Gene expression — Robustness statement 3

The mathematical model with the mRNA transcript and protein dynamics, given in equation (2.33), captures the full CRN model dynamics under the assumption of unlimited ribosomal resources and fast binding reactions. As a result, the error

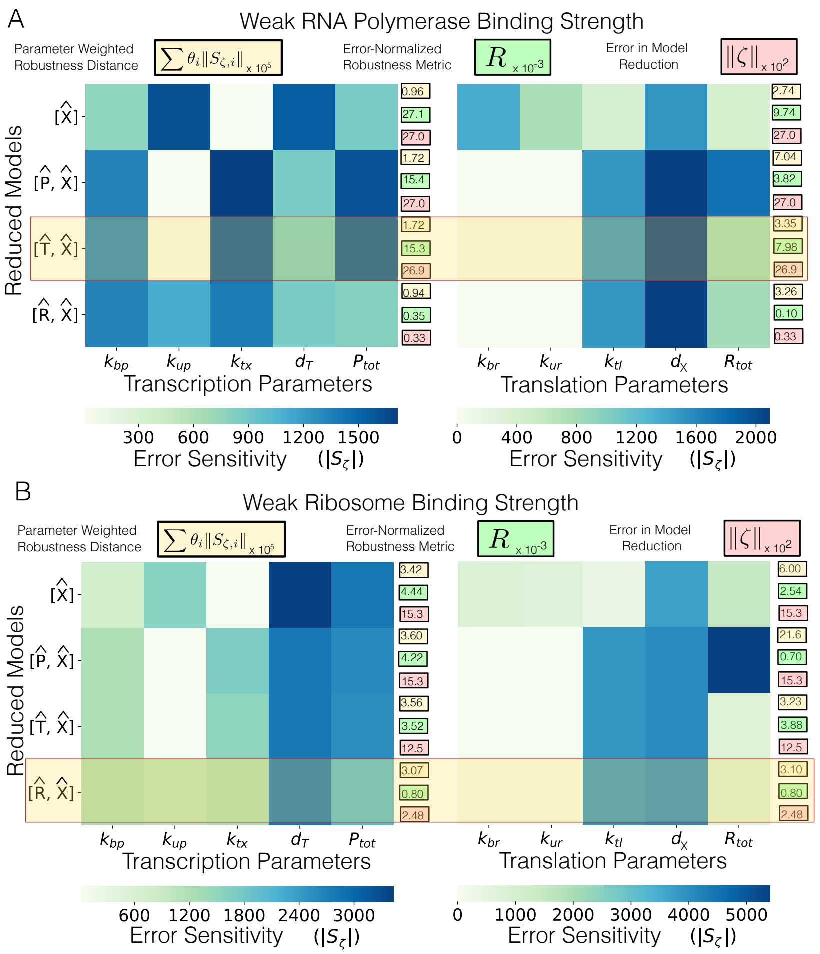


Figure 2.6: Robustness analysis of gene expression models. The figure shows the fragility of different models for all model parameters as computed by the sensitivity of the error in protein X concentration between the full model and the reduced models under different parameter conditions. The weighted robustness distance, the normalized robustness metric, and the normed error in model reduction are labeled for each reduced model in each parameter condition. The combined decision metric r, with weights $w_1 = 1, w_2 = 0.3$, is used to choose one final reduced model in each of the four conditions as shown by the red box. **(A)** Under the condition of weak RNA polymerase binding strength, we get that the mRNA transcript-protein model has the best performance. **(B)** Under the condition of weak ribosome binding strength, we get that the ribosome-protein model ($[\hat{R}, \hat{X}]$) has the best performance as all other models have a much higher error and lower robustness.

performance of the transcript-protein model is directly dependent on the mRNA-ribosome binding/unbinding parameters and the translation parameter. Hence, unlike the $\hat{R}_\Delta$ model, this model is not robust to perturbations in k_{br}, k_{ur}, and R_{tot} in addition to k_{tl} and d_X.

Justification. Similar to the justification of Statement 2, we can analyze the dynamics of the sensitivity of the error to parameter perturbations for the mRNA transcript and protein dynamical model. For the translation rate, that is $\theta_i = k_{tl}$,

$$\dot{S}_\zeta = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial k_{tl}} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial k_{tl}} + \hat{R}_\Delta - \frac{R_{\text{tot}}\hat{T}}{K_0 + \hat{T}} + \frac{k_{tl}R_{\text{tot}}\hat{T}}{k_{br}\left(K_0 + \hat{T}\right)^2} - d_X S_\zeta. \tag{2.40}$$

For the protein degradation parameter $\theta_i = d_X$,

$$\dot{S}_e = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial d_X} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial d_X} - \zeta - d_X S_\zeta. \tag{2.41}$$

For the total ribosome count, $\theta_i = R_{\text{tot}}$, we have,

$$\dot{S}_e = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial R_{\text{tot}}} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial R_{\text{tot}}} - \frac{k_{tl}\hat{T}}{K_0 + \hat{T}} - d_X S_\zeta. \tag{2.42}$$

For the ribsome-transcript binding parameter, $\theta_i = k_{br}$, we have,

$$\dot{S}_\zeta = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial k_{br}} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial k_{br}} - \frac{k_{tl}K_0^2 R_{\text{tot}}\hat{T}}{\left(K_0 + \hat{T}\right)^2 (k_{tl} + k_{ur})} - d_X S_\zeta. \tag{2.43}$$

For the ribsome-transcript unbinding parameter, $\theta_i = k_{ur}$, we have,

$$\dot{S}_\zeta = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial k_{ur}} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial k_{ur}} + \frac{k_{tl}R_{\text{tot}}K_0\hat{T}}{\left(K_0 + \hat{T}\right)^2 (k_{tl} + k_{ur})} - d_X S_\zeta. \tag{2.44}$$

For all other parameters, we have,

$$\dot{S}_\zeta = k_{tl}\frac{\partial \hat{R}_\Delta}{\partial k_{ur}} - \frac{k_{tl}R_{\text{tot}}K_0}{\left(K_0 + \hat{T}\right)^2}\frac{\partial \hat{T}}{\partial k_{ur}} - d_X S_\zeta. \tag{2.45}$$

From equations (2.40) – (2.45), we can conclude that S_ζ directly depends on the total ribosome count, R_{tot}, the binding/unbinding parameters of the transcript with ribosome, k_{br}, k_{ur}, the translation rate, k_{tl}, and the protein degradation rate, d_X. This is in contrast with the results for the $\hat{R}_\Delta$ model where the dynamics of S_ζ only depend on the translation rate and the protein degradation rate. Hence, the $\hat{R}_\Delta$ model is robust to all other parameters whereas the transcript-protein model is fragile to all of these parameters, proving the statement assertion. □

We scale up the above analysis by using a gene expression model that consists of endonuclease mediated mRNA degradation as well.

Gene expression with endonuclease mediated mRNA degradation

To investigate the scalability of this approach, we expand on the gene expression example by including enzymatic degradation of the mRNA transcript mediated by endonucleases. Here we have this enzymatic degradation in addition to the basal degradation rate d_T. The CRN and the corresponding mass-action ODE model is given in Table 2.2.

Table 2.2: Gene expression with endonuclease mediated mRNA degradation

CRN	ODE model	Nominal Parameter Values
$G + P \underset{k_{up}}{\overset{k_{bp}}{\rightleftharpoons}} C_1$	$\frac{dP}{dt} = \left(k_{up} + k_{tx}\right) C_1 - k_{bp} G P$	$k_{bp} = 80, k_{up} = 2$
$C_1 \xrightarrow{k_{tx}} G + P + T$	$\frac{dC_1}{dt} = k_{bp} G P - \left(k_u p + k_t x\right) C_1$	$k_{tx} = 0.5$
$T + R \underset{k_{ur}}{\overset{k_{br}}{\rightleftharpoons}} C_2$	$\frac{dT}{dt} = k_{tx} C_1 + \left(k_{ur} + k_{tl}\right) C_2$	
	$-k_{br} T R + k_{ue} C_3 - k_{be} T E - d_T T$	$k_{br} = 80, k_{ur} = 2$
$C_2 \xrightarrow{k_{tl}} T + R + X$	$\frac{dR}{dt} = \left(k_{ur} + k_{tl}\right) C_2 - k_{br} T R$	$k_{tl} = 0.5$
$T + E \underset{k_{ue}}{\overset{k_{be}}{\rightleftharpoons}} C_3$	$\frac{dC_2}{dt} = k_{br} T R - \left(k_{ur} + k_{tl}\right) C_2$	$k_{be} = 10, k_{ue} = 2$
$C_3 \xrightarrow{d_E} E$	$\frac{dE}{dt} = (k_{ui} + d_E) C_3 - k_{be} T E$	$d_E = 0.1$
$T \xrightarrow{d_T} \varnothing$	$\frac{dC_3}{dt} = k_{be} T E - (k_{ue} + d_E) C_3$	$d_T = 0.5$
$X \xrightarrow{d} \varnothing$	$\frac{dX}{dt} = k_{tl} C_2 - d_X X$	$d_X = 0.01$

Observe that in this model we have the following conservation law relationships,

$$\frac{dP}{dt} + \frac{dC_1}{dt} = 0, \quad \frac{dR}{dt} + \frac{dC_2}{dt} = 0, \quad \frac{dE}{dt} + \frac{dC_3}{dt} = 0.$$

Hence, for constants P_{tot}, R_{tot}, and E_{tot}, we can write

$$P + C_1 = P_{\text{tot}}, \;\; R + C_2 = R_{\text{tot}}, \;\; E + C_3 = E_{\text{tot}}.$$

Using these algebraic relationships, we can eliminate the complexes C_1, C_2, and C_3 to obtain a reduced ODE model. Next, we explore various time-scale separation assumptions that may be used to get further reduced models and discuss their performance and robustness with respect to the model parameters.

The mRNA transcript and protein dynamical model

Assuming that the dynamics of all species in the model other than the mRNA transcript T and the protein X are at quasi-steady-state, we get the following model. Observe that a new degradation Hill function term appears in the dynamics of the mRNA transcript that is dependent on the endonuclease binding parameters

$$\frac{d\hat{T}}{dt} = k_{tx}P_{\text{tot}}\frac{G}{K_1 + K_1G} - d_E E_{\text{tot}}\frac{\hat{T}}{K_2 + \hat{T}} - d_T\hat{T}$$

$$\frac{d\hat{X}}{dt} = k_{tl}R_{\text{tot}}\frac{\hat{T}}{K_0 + \hat{T}} - d_X\hat{X} \tag{2.46}$$

where K_2 is a new lumped parameter that is the Hill activation parameter for the endonuclease binding,

$$K_2 = \frac{d_E + k_{ue}}{k_{be}}.$$

Modeling the dynamics of available free ribosomes

As before, we define the available free ribosomes in the system as $\hat{R}_\Delta = R_{\text{tot}} - \hat{R}$. Using this definition and assuming that the dynamics of all species in the model other than the free ribosomes, the mRNA transcript, and the protein are at quasi-steady-state we get the following model.

$$\frac{d\hat{T}}{dt} = k_{tx}P_{\text{tot}}\frac{G}{K_1 + G} + (k_{tl} + k_{ur})\hat{R}_\Delta - d_E E_{\text{tot}}\frac{\hat{T}}{K_2 + \hat{T}} - k_{br}(R_{\text{tot}} - \hat{R}_\Delta)\hat{T} - d_T\hat{T}$$

$$\frac{d\hat{R}_\Delta}{dt} = k_{br}(R_{\text{tot}} - \hat{R}_\Delta)\hat{T} - (k_{tl} + k_{ur})\hat{R}_\Delta, \quad \frac{d\hat{X}}{dt} = k_{tl}\hat{R}_\Delta - d_X\hat{X} \tag{2.47}$$

Note that the model with only $\hat{R}_\Delta$ and the protein $\hat{X}$ dynamics does not work in this case since the enzymatic degradation reactions for the mRNA transcript are significant for the overall dynamics. So, either T or E is necessary in the ribosome and protein model to get satisfactory performance. As a result, we can also obtain a reduced model with $\begin{bmatrix}\hat{R} & \hat{E} & \hat{X}\end{bmatrix}$ as the states. The performance of all of the possible reduced models is shown in Figures 2.7 and 2.8.

Although a similar robustness analysis as shown in Statements 2 and 3 can be done for this system dynamics, it would be easier to numerically compute a bound on S_ζ for all reduced models. The heatmap of the Euclidean norm of this metric is shown in Figure 2.9. We use the upper bound for the norm of S_ζ to compute the

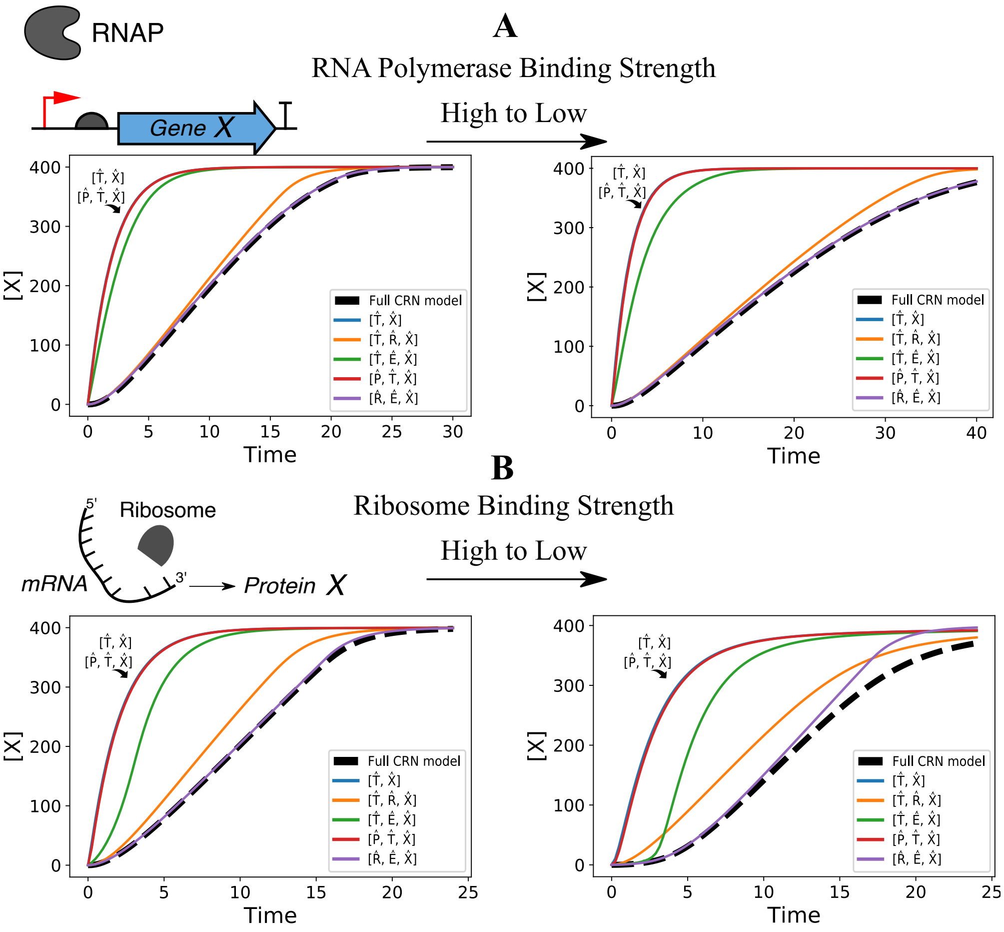
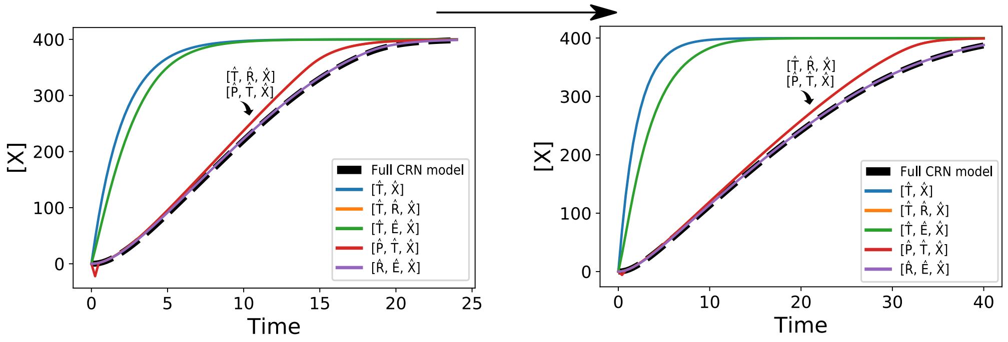


Figure 2.7: Performance of the gene expression with endonuclease mediated mRNA degradation models under different biological conditions. **(A)** We observe that for weaker binding of RNA polymerase to the promoter region of the DNA implies a slower time-response which is only captured perfectly by the free ribosome model that also models the endonuclease dynamics. The mathematical model with only the mRNA transcript and protein dynamics is unable to capture this effect since this binding reaction is assumed to be at quasi-steady-state in this model, but it is the only two-state model for this system with a satisfactory steady-state performance. **(B)** With decreasing ribosome binding strength, none of the reduced models perfectly capture the full CRN dynamics, but still, the ribosome-protein model with endonuclease dynamics ($[\hat{R}, \hat{E}, \hat{X})$ is the closest in error performance to the full model.

robustness estimate for each model parameter. The `AutoReduce` software can be used to compute the model equations and these metrics.

Population control — Non-identifiability analysis

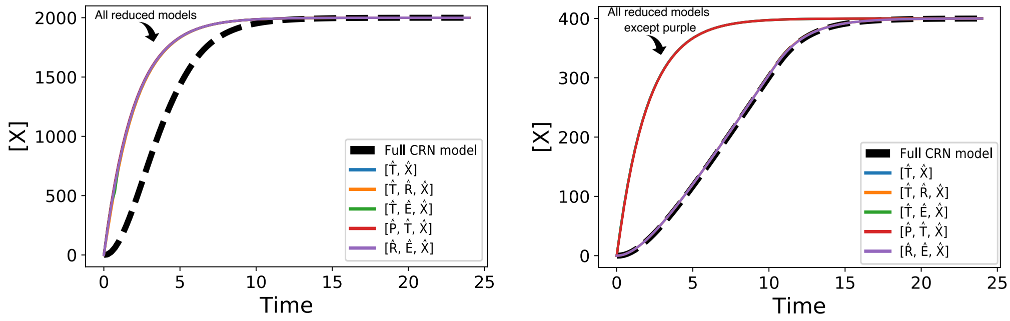
We discuss a population control synthetic biological circuit example to demonstrate the application of model reduction to improve parameter inference. Using this



A

Cellular Resources (Total RNA Polymerase)

High to Low



B

Unlimited Resources

C

Weak Endonuclease Binding

Figure 2.8: Performance of the gene expression with endonuclease mediated mRNA degradation models under different biological conditions. **(A)** We observe a similar effect with decreasing total RNA polymerase in the system. **(B)** With a very high amount of cellular resources, all models reach steady-state faster. **(C)** Under weak endonuclease binding, we observe that only the model that explicitly models the endonuclease dynamics gives good performance.

example, we also show how our approach can be used to analytically obtain the non-identifiable manifold description for a system. We consider a synthetic circuit that controls the population density and composition of a two-member bacterial consortium [7], [102]. In this circuit, under the control of two inducer input signals, each cell kills itself by expressing a toxin protein (ccdB). Each cell type rescues the other cell type by producing a signal that activates the expression of an anti-toxin (ccdA) in its partner cell. There are two different fluorescent signal readouts corresponding to the population of each cell type, given by L_1 and L_2 in the model. The circuit schematic is shown in Figure 2.10-A, B. For the mathematical model, we denote the average concentration of the toxin protein (ccdB) in the cell population with N_i, $i = 1, 2$. Similarly, the anti-toxin (ccdA) is denoted by A_i. Finally, the

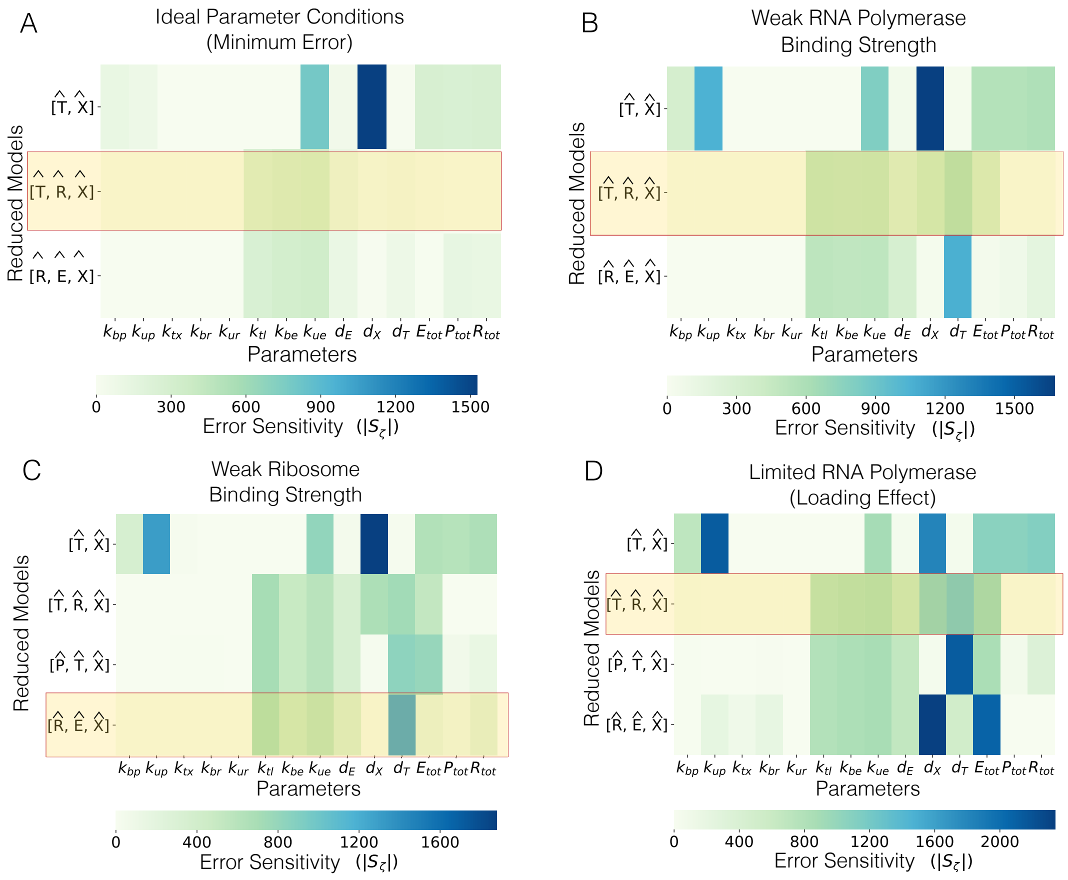


Figure 2.9: Robustness analysis of gene expression models with endonuclease mediated degradation of mRNA. The heatmaps show the norm of S_ζ for each model parameter. Lower S_ζ implies better robust performance. **(A)**: Robustness of the reduced models under ideal parameter conditions corresponding to unlimited resources as shown in Figure 2.8-B where the error for all reduced models is the minimum. Using the combined decision metric $r = w_1 \|\zeta\| + w_2 d_R$, we get that the reduced model with the states $\hat{T}, \hat{R}, \hat{X}$ is the best choice. **(B)**: For weak RNA polymerase binding strength, we obtain that the $\hat{T}, \hat{R}, \hat{X}$ model performs better compared to the other reduced models. **(C)**: Under weak ribosome binding strength condition, we get that the reduced model consisting of $\hat{R}, \hat{E}, \hat{X}$ as its states has the best performance. **(D)**: Under limited resources, the $\hat{T}, \hat{R}, \hat{X}$ model is again able to capture the full CRN dynamics with the least error and most robustness.

concentration of AHL signals in the consortia is given by R_i for each signal. The description of the model parameters and their values are given in Table 2.3. For more details on the parameter values and system description, the reader is referred to [7].

In this mathematical model, an intrinsic assumption is that the signal transport is instantaneous as the dynamics of transport of signals across cell membranes is not explicitly modeled. The inducer signals, denoted by L and I, are the two inputs to the system. The mathematical model is given by the following ODEs:

$$\begin{aligned}
\frac{dN_1}{dt} &= \beta_{R_1}\left(l_{R_1} + \frac{R_1^2}{K_{R_1} + R_1^2}\right) - k_b A_1 N_1 - d_T N_1 \\
\frac{dA_1}{dt} &= K_r \beta_{R_2}\left(l_{R_2} + \frac{R_2^2}{K_{R_2} + R_2^2}\right) - k_b A_1 N_1 - d_T A_1 \\
\frac{dR_1}{dt} &= \beta_{tac}\left(l_{tac} + \frac{I^2}{K_{tac} + I^2}\right) L_1 - d_S R_1 \\
\frac{dR_2}{dt} &= \beta_{sal}\left(l_{sal} + \frac{L^2}{K_{sal} + L^2}\right) L_2 - d_S R_2 \\
\frac{dN_2}{dt} &= \beta_{R_2}\left(l_{R_2} + \frac{R_2^2}{K_{R_2} + R_2^2}\right) - k_b A_2 N_2 - d_T N_2 \\
\frac{dA_2}{dt} &= K_r \beta_{R_1}\left(l_{R_1} + \frac{R_1^2}{K_{R_1} + R_1^2}\right) - k_b A_2 N_2 - d_T A_2 \\
\frac{dL_1}{dt} &= k_C\left(1 - \frac{L_1 + L_2}{C_{max}}\right) L_1 - d_c L_1 \frac{N_1}{K_{tox} + N_1} - dL_1 \\
\frac{dL_2}{dt} &= k_C\left(1 - \frac{L_1 + L_2}{C_{max}}\right) L_2 - d_c L_2 \frac{N_2}{K_{tox} + N_2} - dL_2.
\end{aligned} \tag{2.48}$$

We cannot use singular perturbation theory to reduce this system model since it is not clear how this model can be expressed in the standard form for singular perturbation framework. Using our automated model reduction method, we obtain various possible reduced models for this system and choose the "best" reduced model based on the performance metrics we discussed. Since the full model has two output variables (L_1 and L_2), any reduced model for this system must have these two output variables so that the reduced model may be used effectively for parameter inference.

With our approach, we obtain four different reduced models each with four states. Using the performance metrics discussed earlier, we choose the reduced-order model in equation (2.49) since it is the most robust reduced-model (with the least robustness metric for the given parameters). The advantages of using this reduced model for parameter identification are clear since the reduced models only have 13 parameters (compared to 24 in the full model). Further, since an analytical mapping between the full and the reduced models is available, we can determine the non-identifiable manifold that could assist in the identifiability analysis for this model [47].

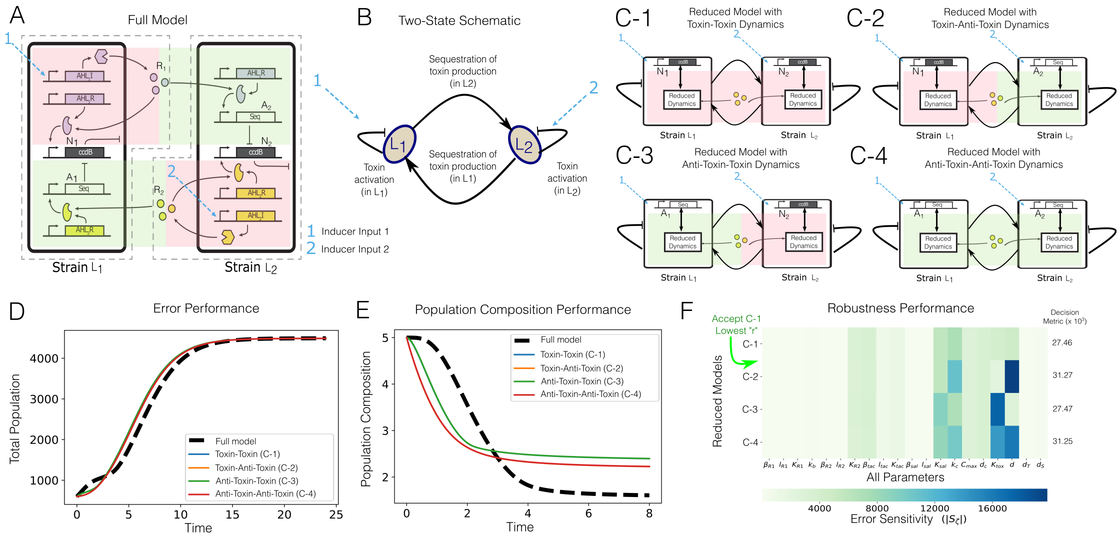


Figure 2.10: Model reduction for population and composition control of a bacterial consortium. **(A)**: Two-member population and composition control circuit schematic. The two inducer inputs (labeled 1 and 2 in cyan) activate the ccdB expression in L_1 and L_2 cell strains, respectively. AHL signals diffuse out of each cell type to signal the expression of anti-toxin (ccdA) in the other cell type. The anti-toxin sequesters away the toxin protein to rescue the cell population. **(B)**: A simple schematic demonstrating the proliferation and death of each cell type under input signals. **(C-1 – C-4)**: Reduced models obtained using our automated model reduction approach. Each reduced model has four states. Two of these states are the output signals which are never reduced in our method. The other two states in each reduced model are labeled in the Figure. **(D)**: Total population ($L_1 + L_2$) obtained on simulating each reduced model and the full model. As we can see, the error performance for all of these four reduced models is satisfactory. **(E)**: Population composition (L_1/L_2) for the reduced models and the full model. Composition control is a feature of this circuit that the reduced models demonstrate as well, although the dynamics of population composition are not fully captured. **(F)**: Robustness analysis for each reduced model gives us a guide to choose a particular reduced model given the parameters in the full model. The heatmap shows the norm of the sensitivity of error in model reduction $\left\|S_{\zeta}\right\|$ with respect to the model parameters. The decision metric $r = w_1 \left\|\zeta\right\| + w_2 d_R$ is then used to choose the reduced model given in (C-1) since it has the least r.

The error and robustness performance for all the reduced models is given in Figure 2.10-D–F. The decision metric r with weights $w_1 = 1, w_2 = 0.5$, is lowest for the reduced model with both of the toxin-states. Hence, we choose this reduced model.

The dynamics for this model are given by:

$$\begin{aligned}
\hat{f}_1 &= \left(l_{R_1} + \frac{\beta_{R_1} x_7^2}{x_7^2 + K_{I0}}\right) - d_T x_1 - \frac{\beta_{R_2} k_b x_1 \left(K_{a0} l_{R_2} + x_8^2\right)}{k_b x_1 x_8^2 + K_{a0} k_b x_1 + d_T x_8^2 + K_{a0} d_T}, \\
\hat{f}_2 &= \left(l_{R_2} + \frac{\beta_{R_2} x_8^2}{x_7^2 + K_{a0}}\right) - d_T x_5 - \frac{\beta_{R_1} k_b x_5 \left(K_{I0} l_{R_1} + x_7^2\right)}{k_b x_5 x_7^2 + K_{I0} k_b x_5 + d_T x_7^2 + K_{I0} d_T}, \\
\hat{f}_3 &= k_c \left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_7 - \frac{d_c x_1 x_7}{x_1 + K_{tox}} - d x_7, \\
\hat{f}_4 &= k_c \left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_8 - \frac{d_c x_5 x_8}{x_5 + K_{tox}} - d x_8.
\end{aligned} \tag{2.49}$$

Here $\hat{x} = \begin{bmatrix} x_1 & x_5 & x_7 & x_8 \end{bmatrix}^T$ and the exact expressions for the lumped parameters are available part of our software package, AutoReduce, on GitHub [88].

2.6 Conclusion

Our main result in this chapter gives a closed-form expression for the robustness guarantee of structured model reduction of linear dynamical systems. We show two different methods to derive this result — a direct linear analysis approach for the linear systems and a sensitivity analysis based approach that also works for nonlinear dynamics. The advantage of our method is that the system does not need to be in the standard form as in singular perturbation theory and that we can compute the robustness of the model reduction error with respect to each model parameter individually for a holistic analysis of different possible model reductions.

We demonstrated the applications of our model reduction approach to biological system examples. The exploration of design space by changing experimentally tunable parameters in the models as shown in Figures 2.2 and 2.3 is an important step towards using mathematical models for biological circuit design. There are two main advantages to this approach:

1. For any complex biological circuit, a single gene expression would usually form a small part of the design. Hence, interpreting the key states and parameters involved in tuning its dynamics is important for the modular design of the complete circuit.

2. For parameter identification, reduced models are commonly used but it is important to verify the correctness of these reduced models. The automated

model reduction method discussed in this chapter is a step towards that goal as it provides a mapping of the reduced models with the full model alongside the performance guarantees. Further, due to parameter lumping, the reduced models have fewer number of parameters, improving the parameter identifiability of the system given measurement data [46], [47]. The non-identifiable manifolds are analytically known as well.

Table 2.3: Model parameters

S.no.	Parameters	Description	Unit	Guess
1	β_{R_1}	Max transcription rate of inducible promoter (for N_1 and A_2)	con./hr	6
2	l_{R_1}	Leak constant of inducible promoter (for N_1 and A_2)	N/A	2e-3
3	K_{R_1}	Activation constant of inducible promoter (for N_1 and A_2)	con.	430
4	k_b	Binding rate between toxin and anti-toxin	1/con.hr	30
5	β_{R_2}	Max transcription rate of inducible promoter (for N_2 and A_1)	con./hr	6
6	l_{R_2}	Leak constant of inducible promoter (for N_2 and A_1)	N/A	2e-3
7	K_{R_2}	Activation constant of inducible promoter (for N_2 and A_1)	con.	190
8	β_{tac}	Max transcription rate of inducible promoter (for R_1)	con./hr	19.8e-3
9	l_{tac}	Leak constant of inducible promoter (for R_1)	N/A	1.5e-3
10	K_{tac}	Activation constant of inducible promoter (for R_1)	con.	1.4e5
11	β_{sal}	Max transcription rate of inducible promoter (for R_2)	con./hr	14.4e-3
12	l_{sal}	Leak constant of inducible promoter (for R_2)	N/A	2.1e-4
13	K_{sal}	Activation constant of inducible promoter (for R_2)	con.	13
14	k_C	Cell division rate	1/hr	0.6
15	C_{max}	Population cap	conc.	5500
16	d_c	Cell death rate	1/conc.hr	0.8
18	I	Max induced I (input 1) concentration	con.	1e6
20	L	Max induced L (input 2) concentration	con.	324
21	K_{tox}	Repression coefficient of toxin to proliferation	con.	1
22	d_S	Degradation constant of AHL signals	1/hr	0.5
23	d	Basal degradation of each cell	1/hr	0.1
24	d_T	Basal degradation of toxins and antitoxins	1/hr	1.5
25	K_r	Ribosome scaling factor	N/A	5

Chapter 3

A BIOLOGICAL MODELING, ANALYSIS, AND LEARNING PIPELINE

3.1 Introduction

We aim to use the reduced-order modeling framework discussed in Chapter 2 to guide the design of a broader class of biological system models. Towards that end, we need an automated method to build detailed chemical reaction network models and then use the reduced model in a parameter inference pipeline to learn and predict system dynamics.

In this chapter, we develop a modeling, analysis, and parameter identification pipeline to guide the design of biological systems. To create this pipeline, we build on three Python tools:

1. BioCRNpyler, for compiling chemical reaction network models at various levels of detail and then switching mechanisms "ON" or "OFF" systematically to explore modeling hypotheses,
2. AutoReduce, for dimensionality reduction of models, conservation analysis of CRNs, and metrics for goodness of reduced models, and
3. Bioscrape, for simulations and parameter identification. We extend the Bioscrape package by developing an easy-to-use parameter identification wrapper around the Python emcee software. The Bioscrape inference package can also be used to validate models using distributional data from flow cytometry, and end-point data under multiple conditions.

We demonstrate this pipeline by characterizing two DNA recombination enzymes called integrases and excisionases in cell-free protein expression system. We systematically obtain a hierarchy of mathematical models for this system using these tools. Then, we use experimental data to validate our mathematical models. Finally, with validated mathematical models, we guide the experimental design choices (such as the ratio of integrase and excisionase enzyme). In this way, we are able to accurately quantify the enzyme action in cell-free systems.

A review of biological modeling tools

To apply the mathematical modeling framework developed in the previous chapter, it is crucial to integrate it with other tools that can simulate the models and infer parameters.

Over the past few years, there has been widespread adoption of software tools in synthetic biology research for modeling, simulation, analysis, data exchange, and design optimizations. The focus on bio-design automation and rational design in synthetic biology has led to this enthusiastic acceptance of software tools. A few examples include Synthetic Biology Open Language (SBOL)/Systems Biology Markup Language (SBML) compatible tools (for data and model standardization) [90], [103], COPASI [30] (for modeling and simulation), iBioSim [28] (for CAD-style circuit modeling), Tellurium [29] (for text scripting of circuit models), promoter/RBS calculators [104], [105] (for prediction of transcription/translation initiation rates), and automated design recommendation tools [106], [107].

The rise in tools for specific tasks has led to their integration into pipelines and repositories like SynBioHub [108], Galaxy SynBioCAD [109], and Infobiotics [110]. SynBioHub is a platform that facilitates the integration of software and data for synthetic biological designs so that users can easily share and reproduce system designs in a standardized format. Similarly, an end-to-end metabolic design portal called Galaxy SynBioCAD chains tools together into various workflows for common design and analysis tasks such as genetic design, flux balance analysis, and pathway benchmarking. While SynBioHub is focused on the standardization of software and data used in synthetic biology pipelines, Galaxy SynBioCAD allows for automated engineering and analysis workflows for designing metabolic pathways. A similar experimental design automation software that focuses on the test and learn parts of the design-build-test-learn cycle called RoundTrip [111] has been developed recently.

An important aspect of rational synthetic biology design that is not addressed by these approaches is model-guided design, or forward engineering, where specifications of circuits can be converted to mathematical models with parameters inferred from experimental data. A software suite that claims to integrate modeling, simulation, and verification for synthetic biological circuits was recently presented, called the Infobiotics Workbench [110]. Infobiotics develops a domain-specific language

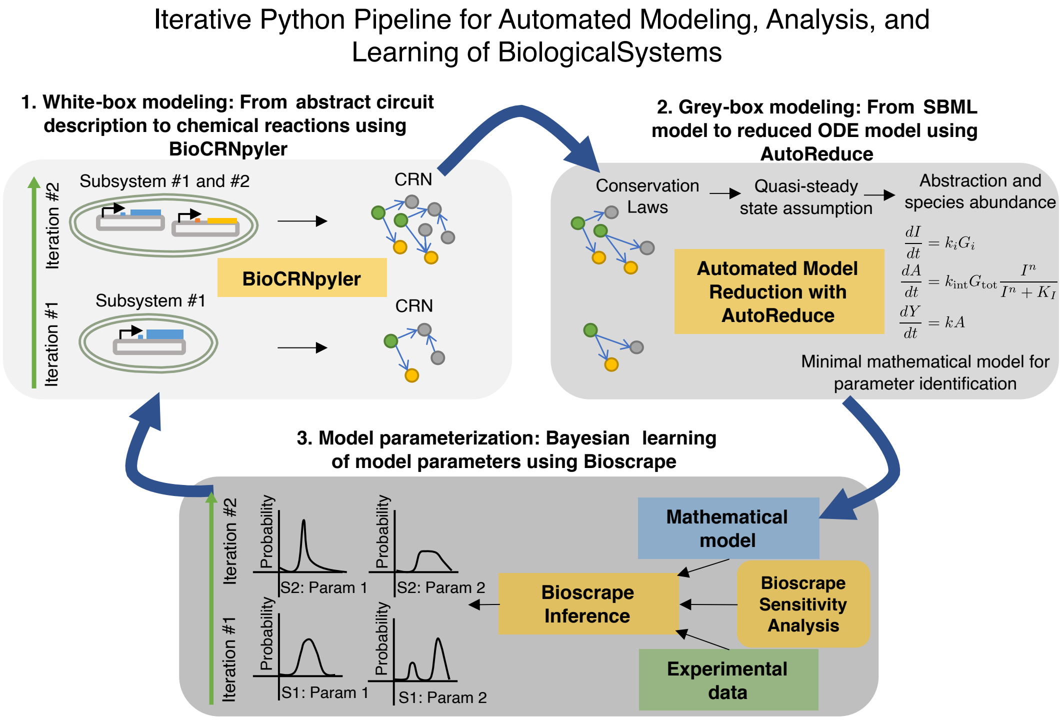


Figure 3.1: An iterative Python pipeline for modeling, analysis, and learning of biological circuits. In the first iteration, we build the CRN model for subsystem #1 and then obtain the minimal representation suitable for parameter identification. Bayesian inference is used to find parameter distributions. The results of the Bayesian inference are fed back into the more detailed model that is built in the second iteration. The previously identified context and circuit parameters are fixed in this larger model so that the analysis can now focus on the new parts introduced in the circuit. For each iteration, a CRN model is compiled using BioCRNpyler, this is the white-box modeling step that includes the mechanistic details of the system. With AutoReduce, we obtain a hierarchy of lower dimensional ODE models under different modeling assumptions. We call this grey-box modeling because we can tune the granularity of the models. Finally, we validate the model parameters in the reduced model by using Bioscrape.

for synthetic biology circuits to write specifications and provides a Java-based GUI platform for modeling and analyses. It is clear from these recent efforts that, for scalable biological circuit design, there is a need for open-source, user-friendly, and easy-to-access modeling and analysis pipelines that work alongside experimental data in a design-build-test-learn cycle.

Towards that end, we present an automated Python pipeline for iterative modeling, model reduction, analysis, and parameter identification of synthetic biological circuits. We further develop the Python software packages — BioCRNpyler [26] (to build models), AutoReduce [112] (to obtain reduced models), and Bioscrape [31] (for simulations, analysis, and Bayesian inference using Emcee [113]) — to create the computational framework shown in Figure 3.1.

A key idea with the proposed pipeline is its iterative nature by breaking down the system analysis into smaller parts. Once we learn the parameters of subsystem #1 and the system context, we fix these parameters for the next iteration. In this way, the parameterization of the model with both subsystem #1 and #2 is feasible and leads to reliable predictions that can guide the experimental design for the bigger system. This process of system identification and learning by parts to guide the design of more complex circuits can be extremely important for scalable biological circuit design and analysis. The proposed pipeline is a step in this direction. Further, the pipeline uses standardized data and model formats to allow for interoperability of tools and integration with other existing pipelines. To demonstrate the tools, we apply this pipeline to characterize an integrase and excisionase-mediated DNA recombination circuit in a cell-free extract.

A short review of DNA recombination circuits

Recombinase-based circuits are circuits that utilize the unique functionality of phage integrases. Phage integrases are responsible for catalyzing the site-specific insertion of bacteriophage DNA into a host genome. These enzymes utilize specific attachment sites found in both the phage and host DNA: attP and attB, respectively. Upon integration of the phage DNA into the host genome, the attachment sites attL and attR are formed [114]. This insertion can then be reversed by the expression of a similar enzyme called, excisionase. Excisionases work in conjunction with the integrases to excise the phage DNA from the host DNA, restoring the original attB and attP sites [115]. Further research and characterization of these enzymes led to the discovery that the presence of anti-parallel attB and attP sites in the same strand of DNA leads to the inversion of the flanked segment, generating anti-parallel attL and attR sites. This attL and attR flanked DNA segment can then be reverted to its original orientation by an integrase-excisionase complex [116].

To date, recombinase-based circuits with various functionality and applications have been designed and experimentally validated. For example, event ordering detection circuits [117], gene networks that count [118], and Boolean logic and memory circuits [119] have been created in living cells using recombinases. Although there has been progress in exploring the mechanisms for enzyme-mediated DNA recombination, the dynamics and characteristics of integrases and excisionases have not yet been fully quantified. In order to accelerate the design and implementation of increasingly complex recombinase circuits, novel methods are required to efficiently quantify their complex dynamics.

We used the proposed modeling and analysis pipeline to study the dynamical models of integrases and excisionases. We anticipate our pipeline providing a foundation to guide further design and implementation of increasingly complex recombinase circuits. In applying this pipeline, we built mathematical models and validated the mechanistic parameters for Bxb1 – a well-characterized and studied serine integrase-excisionase system [120]–[122]. This pipeline, however, can be used to characterize general biological systems to guide the design of progressively complex circuits.

For the demonstrative example of DNA recombination circuits:

1. We develop new chemical reaction network (CRN) mechanisms in BioCRNpyler for integrase action to flip a promoter and the excisionase action to reverse it. These mechanisms model the details of the intermediate complexes and the binding rates of integrase to the DNA, excisionase to the integrase, the complexes, and to the DNA. With model reduction and *in silico* analysis, we show which mechanisms describe the expected behavior and then validate these models with *in vitro* cell-free data.
2. We show an experimental validation of the mathematical models of the integrase and excisionase mediated DNA recombination circuit in a cell-free system. After compiling, reducing, and parameterizing our model from experimental data, we are able to accurately model the effect of integrase and excisionase gene concentrations on fluorescent reporter output. For example, on adding higher integrase DNA concentrations, our models correctly predict the experimental observation that the reporter expression increases while it decreases as more excisionase is expressed.

3.2 Modeling integrase activity in cell-free systems

To characterize the integrase activity independent of the excisionase, we use a two plasmid system — (1) Bxb1 integrase expressing plasmid fused with CFP to measure

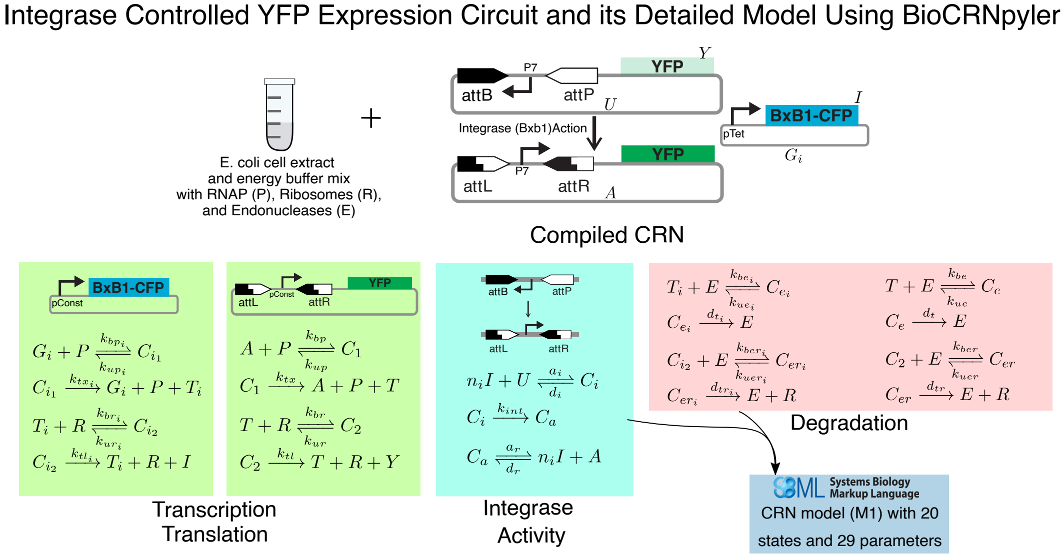


Figure 3.2: Modeling of integrase expression and activity in cell-free systems. The circuit consists of two plasmids, one expressing the Bxb1 integrase and the other with a reversed promoter upstream of the YFP reporter. The integrase activity flips the promoter so that YFP is expressed. BioCRNpyler is used to convert this abstract system description into a CRN model written as an SBML file with 20 species (annotated in the figure) and 29 parameters (details in Table A3 and online). All species definitions in the model are given in the model species section in the Appendix A.

integrase expression, and (2) a YFP plasmid that gets activated on integrase action (shown in Figure 3.2). Using this circuit, we characterize the integrase expression in a cell-free extract and its flipping action on a promoter to control the fluorescent reporter expression.

Towards this first goal, we model the two plasmid system in a cell-free extract using a detailed mechanistic chemical reaction network (CRN) with mass-action kinetics using BioCRNpyler. In BioCRNpyler, a CRN is compiled by specifying the circuit parts as `Component` objects, their interactions as `Mechanism` objects, and the context for the circuit as a `Mixture`. For the integrase circuit, we create an integrase component and two new mechanisms — a simple integrase flipper mechanism with one reaction that converts the inactive DNA to active by flipping the promoter direction and a detailed integrase flipper mechanism that models the binding reactions of the integrase to various DNA sites. Finally, to compile a CRN model, we use an existing BioCRNpyler mixture, `TxTlMixture` to model the cell-free context and resources. The detailed reactions for this model are given in Figure 3.5A. We simulate this CRN model using Bioscrape (shown in Figure 3.3) to explore the

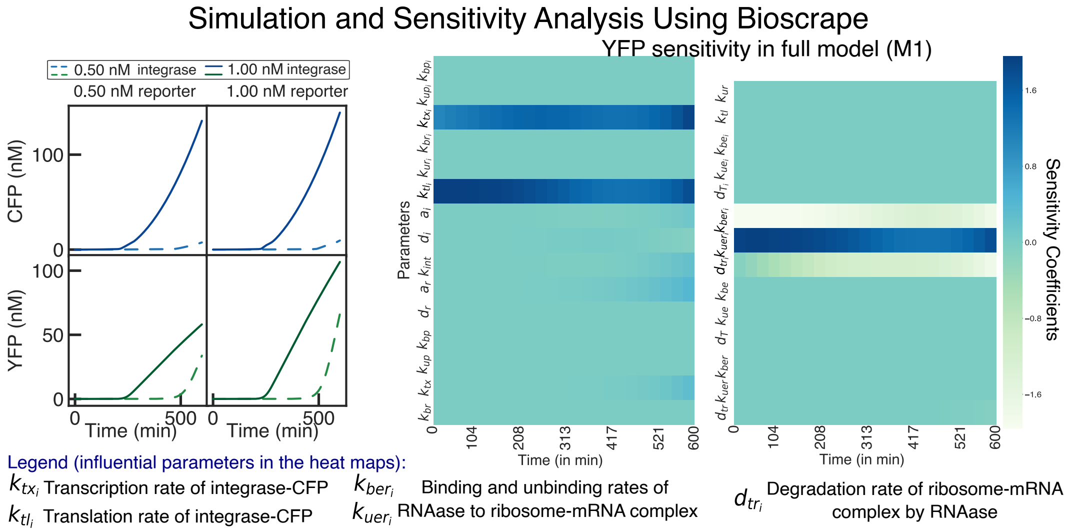


Figure 3.3: Simulation and analysis of integrase expression and activity in cell-free systems using Bioscrape. The simulations predict the CFP and YFP expressions under different initial conditions. The sensitivity analysis shows the most influential parameters for the time course of YFP expression.

design space and the resource-loading effects. However, the detailed model is infeasible to fit the experimental data due to the problem of unidentifiability [123] and high-dimensionality. Hence, we use the methodology described in Chapter 2 of this thesis to automatically derive potential reduced models for this system. First, we derive and apply the conservation laws to reduce the system model. Then, we apply quasi-steady state approximation (QSSA) to obtain reduced models under different assumptions. Using model reduction performance metrics, we choose reduced models that recover the desired properties (integrase flipping, fluorescent reporter levels, and any other important context effects), shown in Figure 3.4 as M-3. The model reduction methods and the related performance metrics are discussed in the Methods section at the end of this chapter. To obtain a further reduced model, we abstract the context by switching off resource-dependent mechanisms for transcription and translation in BioCRNpyler. Then, we further reduce the one-step transcription and translation model using QSSA and assuming the abundance of certain species to obtain a minimal ODE model (M-4). It is evident from Figure 3.4 that the model M-4 recovers the commonly used Hill function, however, no heuristics were used in deriving this model. We export this model as SBML for further analysis.

At each step in the model reduction hierarchy, the number of states and the parameters are reduced. For better accuracy of the reduced model, the user may

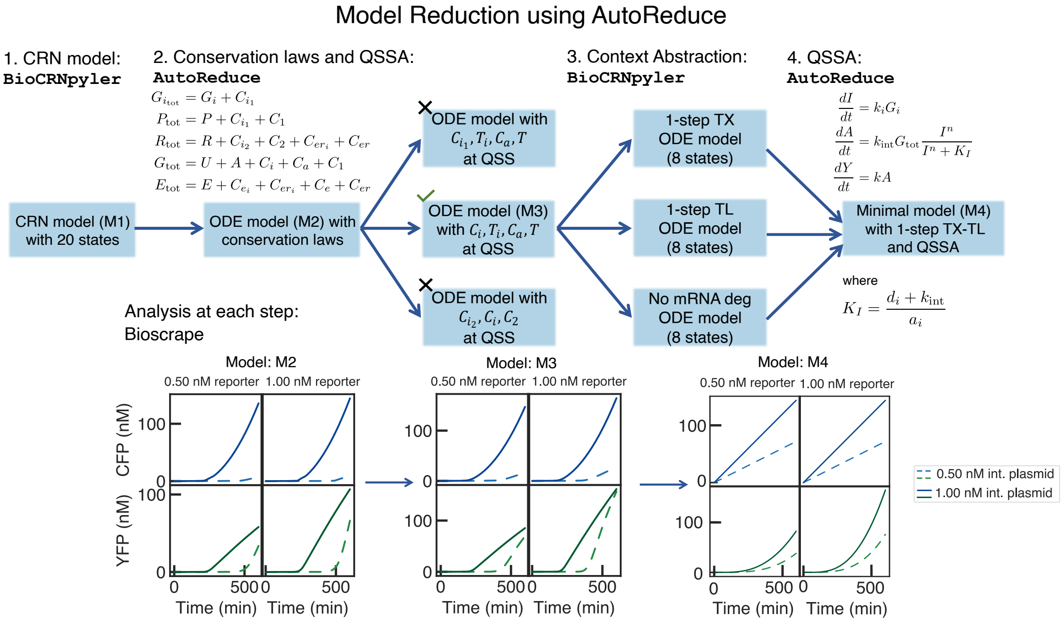


Figure 3.4: A hierarchy of reduced models for integrase expression and activity in cell-free systems using AutoReduce. The CRN model is reduced in multiple steps with AutoReduce. First, the conservation laws are determined (as shown in C-2) and a reduced model is obtained symbolically. This reduced model has 5 fewer states than the full model and is further reduced using QSSA. Multiple reduced models are possible at this stage, out of which one model, M3, is selected (tick marked in the figure) based on the error performance metric as computed by AutoReduce. Then, a minimal model is obtained by abstracting the details and further reducing the model using QSSA and species abundance assumptions. The minimal ODE model (M4) is shown in C-4. The simulations for reduced models (M2, M3, and M4) are also shown.

choose a model that is higher up in the hierarchy, whereas, for faster parameter inference, the minimal model with the fewest parameters may be chosen. The computational run time to obtain each reduced model varies from a few seconds up to a maximum of a few minutes on an i7-6700K Intel CPU laptop with 16GB of RAM.

For the minimal model, we find the identifiable parameters as the most sensitive parameters affecting the measurements. The sensitivity analysis tools in Bioscrape (shown in Figure 3.5B) show the sensitivity of each model parameter with time for each output measurement (CFP and YFP). For the identifiable parameters, we use Bayesian inference tools in Bioscrape to fit the cell-free data (see Figure 3.5B and the inference section in Methods).

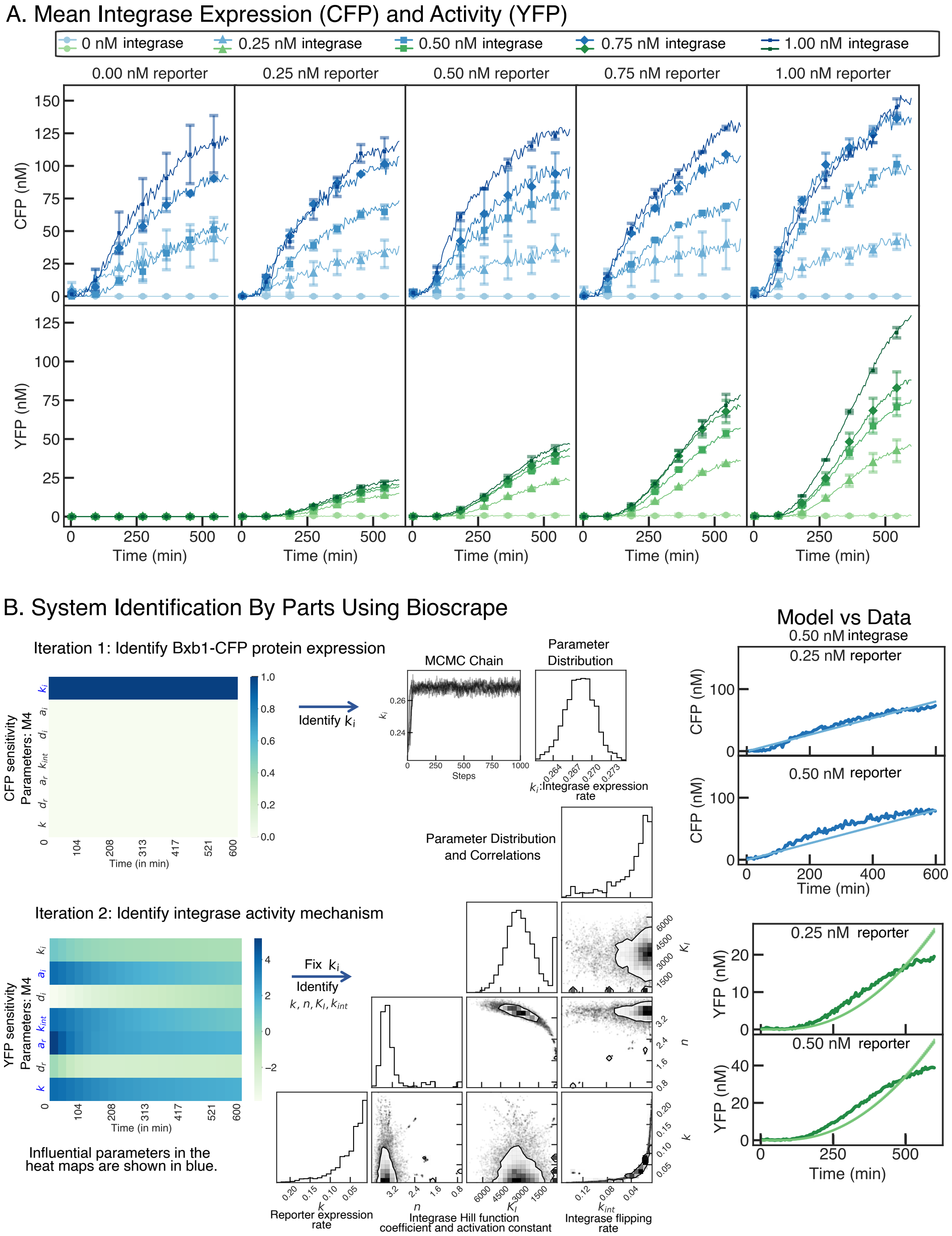


Figure 3.5: Experimental data and system identification by parts of cell-free integrase expression and activity. (A) Mean background-subtracted fluorescence data for the integrase circuit in the cell-free extract. (B) We identify the system by parts, that is, we first select the integrase expression part of the circuit and run sensitivity analysis to find out its identifiable parameters. We observe that k_i is the only sensitive parameter and hence, we run Bayesian inference to identify the posterior parameter distribution for k_i. The model fit alongside the data is shown in the rightmost column. Once we have identified this part, we fix the corresponding parameter, k_i, and run the sensitivity analysis for YFP output. We identify all parameters that YFP is sensitive to. The corner plot shows the posterior distributions of each parameter alongside their correlations with 75% confidence contours. The mismatch in the data and the model is due to the minimal model not capturing the plateauing expression as cell-free extract stops protein expression.

The parameter inference algorithm is implemented in Bioscrape as a plug-and-play Python wrapper for the emcee package that implements a Markov Chain Monte Carlo (MCMC) sampler for Bayesian inference. We import the SBML file for the model and the experimental data as CSV and then run the inference after choosing appropriate MCMC parameters (see Appendix A). After running the sampler, we obtain posterior distributions for the parameters which are then used for plotting the identified model simulation against the experimental data. Hence, the “full-stack” Python pipeline of modeling, design-space exploration, sensitivity analysis, model reduction, and parameter inference gives us a validated mathematical model for cell-free integrase activity.

Modeling excisionase activity in cell-free systems

Integrase action activates the output fluorescence protein expression by recombining the attP-attB site on the DNA to the attL-attR site so that the promoter is flipped towards the protein coding sequence. To accurately control this expression, we use the reverse directionality factor, or the excisionase enzyme, which can reverse the promoter direction by changing from the attL-attR site back to the “inactive” attP-attB site on the DNA. We design a new plasmid that expresses the excisionase fused with mScarlet to measure the excisionase expression (see Figure 3.6).

We build on the integrase models (both detailed and simplified mechanisms) shown in the previous section to develop excisionase mechanisms. We hypothesize that the excisionase can have two possible mechanisms to reverse the integrase activity:

1. The excisionase binds to the integrase and the resulting complex then binds to the attL-attR site on the DNA to flip it to the attP-attB site. The integrase-excisionase complex may also bind to the attP-attB site, sequestering the site from further recombinations.
2. The excisionase binds to the integrase-bound DNA sites to form a complex that then flips the promoter region or keeps it sequestered.

With BioCRNpyler, it is possible to switch `ON` any one or both of these mechanisms when compiling the CRN model. See Figure 3.6 for the list of chemical reactions. We compile a detailed integrase-excisionase model in BioCRNpyler with these mechanisms added into the same `TxTlMixture` as was used for the integrase model. This model consists of 30 species and 54 parameters.

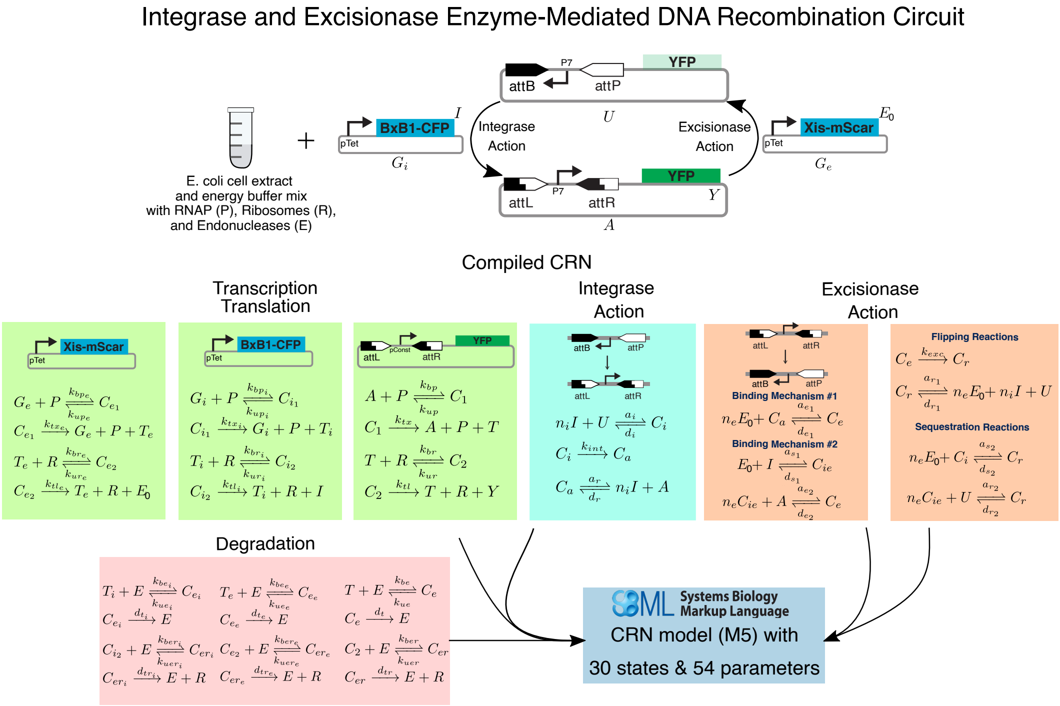


Figure 3.6: CRN models for excisionase expression and activity in cell-free extract using BioCRNpyler. We obtain a CRN for the circuit with both integrase and excisionase in cell-free extract by describing the circuit specifications in BioCRNpyler. The resulting model has 30 species (annotated in the figure) and 54 reaction rate parameters (details in the Table A4 and online). The species definitions are given in Appendix A on model species.

For this model, we predict various parameter values from the characterization of the integrase circuit in cell-free extract. We use the identified and validated parameters for the cell-free extract resources, integrase expression, and integrase action in the excisionase model. The integrase action parameters in this model are context-dependent, so, we allow for these to be updated as we validate using the integrase-excisionase cell-free data. However, we keep the cell-free resource parameters constant as they model the total resources provided by the extract.

We analyze the sensitivities of each parameter in this model to the output and run simulations under various conditions to predict the excisionase action *a priori*. This guides our design choices of choosing the excisionase plasmid levels relative to the integrase plasmid levels. As we vary integrase plasmid initial conditions from 0 to 2 nM, we observe that varying the excisionase initial conditions from 0 to 1 nM gives accurate control of output protein desired levels of expression. This is the design choice we made for the cell-free experiments with both integrase, excisionase, and reporter plasmids (resulting *in vitro* data shown in Figure 3.9). These forward design choices driven by mathematical models were made possible due to the characterization of a smaller part of this circuit (the integrase-reporter circuit), the detailed models, their sensitivity analyses, and with preliminary excisionase-reporter cell-free experiments.

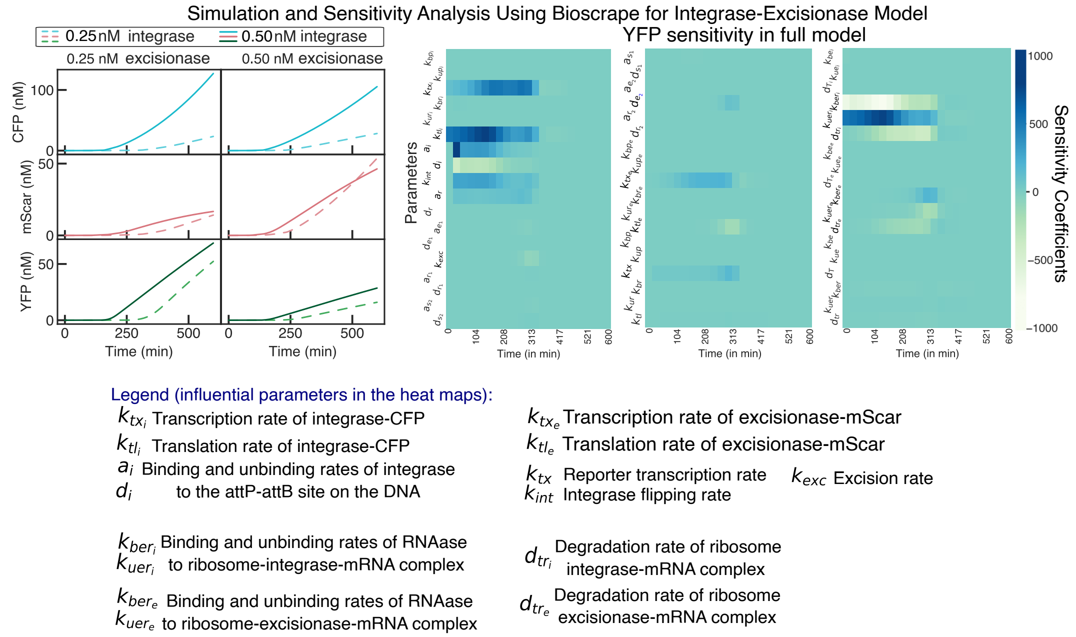


Figure 3.7: Analysis for excisionase expression and activity in cell-free extract using Bioscrape. We show the simulation for the CRN model. In this model, we use the identified integrase parameters to predict the excisionase activity. We observe that as more excisionase is expressed, YFP expression falls down. The sensitivity of each parameter in the CRN model with time is shown in the sensitivity analysis heatmaps.

To validate the mathematical models by identifying parameter values from the experimental data, we need to reduce the dimension of the parameter space of the detailed model. Although the detailed model captures a range of behavior, such as resource loading and competition, it is infeasible for parameter identification due to the unidentifiability of its large parameter space. So, we use AutoReduce to reduce the models in multiple steps.

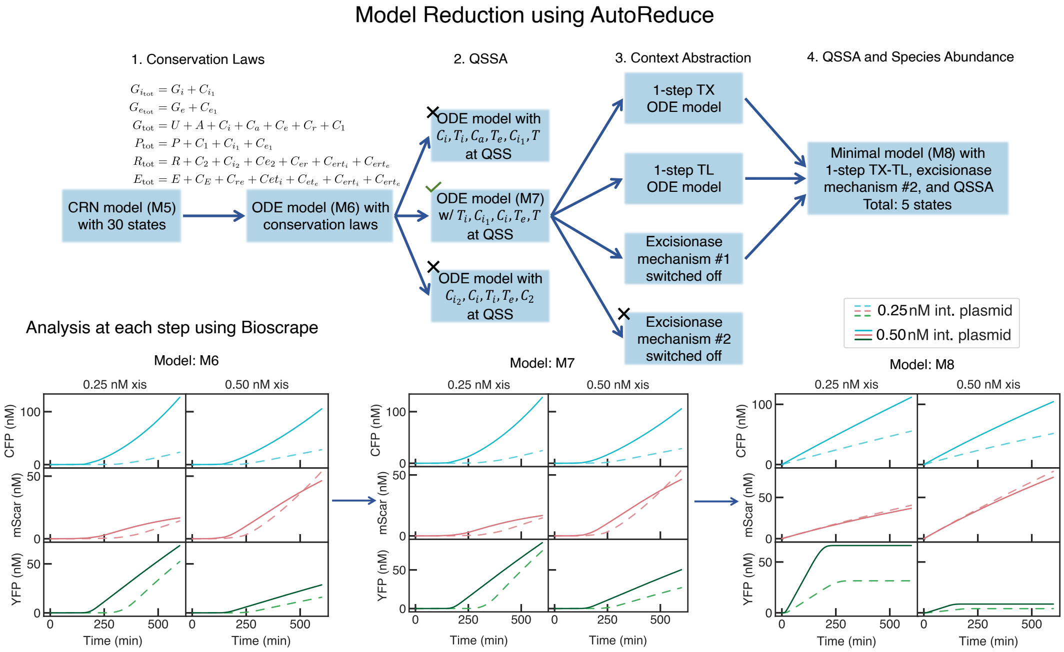


Figure 3.8: Mathematical models for excisionase expression and activity in cell-free extract obtained using AutoReduce and BioCRNpyler. Using AutoReduce, we find the conservation laws in the CRN model given by BioCRNpyler. After applying conservation laws, we further reduce the dimensionality of the integrase-excisionase system by assuming species at a quasi-steady state (QSS). Finally, we abstract the context of resource details modeled using BioCRNpyler and reduce these models further using AutoReduce to obtain a minimal model (M8). The chosen reduced models are marked with a tick while a cross indicates discarded reduced model. The simulations for three reduced models are shown — M6, M7, and M8. The computational run time to derive these reduced models are similar to the integrase reduced models, varying from a few seconds to a few minutes.

This model reduction process is shown in Figure 3.8 which starts at automatically deriving and applying the conservation laws, then proceeds to QSSA, and finally, context abstraction with BioCRNpyler to derive a minimal ODE model (M-8 in Figure 3.8). This minimal model is suitable for parameter identification as it contains only 5 states and 17 parameters out of which 4 parameters are most sensitive to YFP expression (see Figure 3.9B).

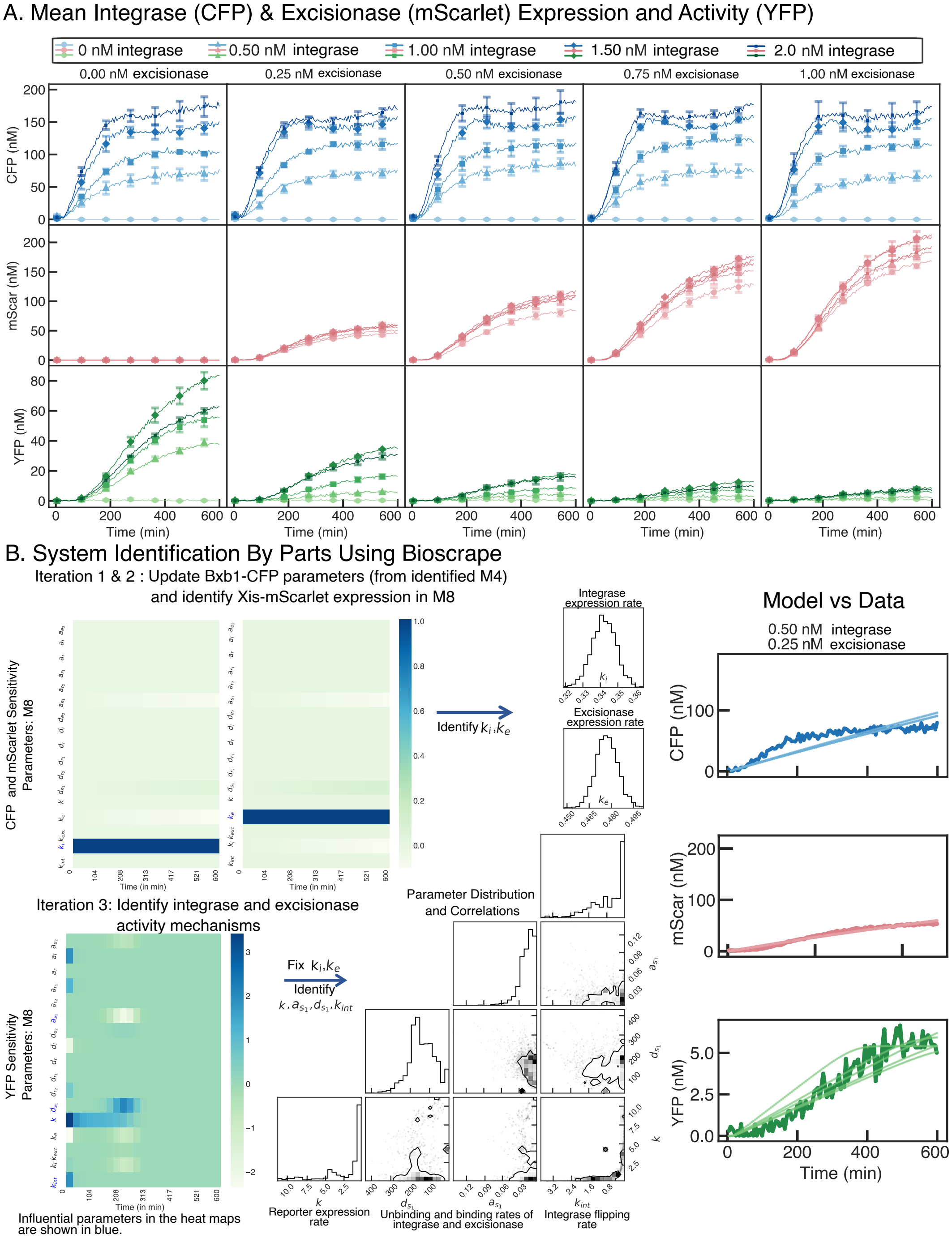


Figure 3.9: Experimental data and system identification by parts of integrase and excisionase expression and activity in cell-free extracts. (A) Mean background-subtracted fluorescence data for the integrase-excisionase circuit in cell-free extract. As more integrase is added, more YFP is expressed until the maximum possible expression is reached at 1.5nM integrase. With higher excisionase levels, we observe a decreasing gradient of YFP levels. (B) To identify the model parameters, we set the previously identified integrase parameters and update accordingly to account for loading effects. In the second iteration, we infer the parameters for the mScarlet expression. Finally, we identify the sensitive set of parameters for YFP. In the right panels, we show the identified parameter distributions and the data plotted alongside the model simulations.

Using the identified parameters, we are able to characterize and predict the excisionase activity in reversing the integrase action, their relative strengths, and the ratios required for accurate protein level prediction. The experimental data and the parameter identification steps for excisionase characterization are shown in Figure 3.9.

3.3 Conclusion

We present a new *in silico* pipeline to assist the design-build-test-learn process in synthetic and systems biology. This pipeline consists of user-friendly, open-source, Python software packages that have been developed with community input and support standardized models written in SBML. We build on three Python packages to formulate the pipeline presented in this chapter, which starts by building detailed reaction network models for biological circuits, reducing the detailed models to simpler ODEs, and seamlessly connecting these models to experimental data through Bayesian inference. All software packages demonstrated in this chapter are Python-based (for easy integration with other tools and pipelines) and have an active discussion and issue support forums through Github.

We demonstrate the application of this pipeline with a novel circuit design in a cell-free protein synthesis system. We build a three-plasmid system consisting of DNA recombination enzymes, integrases, and excisionases. With the help of this pipeline, we characterize the expression and action of integrases and their reverse directionality factors, excisionases. This characterization involves detailed reaction network models, ODE models under various assumptions with a clear indication of when each model is valid, and posterior parameter distributions for model parameters that fit the experimental data.

We quantify the expression strengths of integrase, excisionase, and the reporter while also quantifying their relative effects on the output fluorescence measurement. Particularly, we show control of output protein expression at various levels with experimental data that is backed by mathematical models. We postulate that predictable DNA recombination-based control of protein expression adds an alternative design choice for biological circuit designs. This characterization of circuit parts in the cell-free system with mathematical models would allow for such complex circuit designs that use integrases and excisionases for precise control of protein expression. While further characterization and experimentation, particularly with regard to loading effects, are needed to fully realize the potential of these components, the computational pipeline we have developed is flexible enough to support these future research directions.

To further illustrate the potential for using our characterized parts to design larger circuits, we have provided a Python notebook on our Github [124] repository that contains modeling and extensions of recombination-based circuits discussed in the literature. This notebook includes the modeling and simulation of a recombination-based AND logic gate circuit [125] that can be easily constructed using the characterized CRN mechanisms from this study. We also model a DNA recombination-based finite state machine [125]. These models serve as a useful resource for exploring the design space of DNA recombination-based circuits and allow users to easily build upon the results of our study to create larger circuits using the characterized parts. By making this resource available, we hope to encourage the development of DNA recombination-based circuits and demonstrate the utility of our work.

3.4 Methods

Model reduction methods

We used a variety of model reduction techniques to derive reduced models in this chapter. We chained these methods in an automated workflow by developing the Python model reduction software, AutoReduce [112]. First, all CRN models with mass-action kinetics admit a set of conservation laws. The underlying theory and the derivation of conservation laws is a well-studied topic in CRN theory [126]. We implemented Python methods in AutoReduce to search the conservation laws in a given model (not necessarily a mass-action CRN), computation of reduced models with conservation law substitutions, and symbolic manipulation of the model as well as numerical computations for symbolic models. For CRNs with mass-action kinetics, we find that the derived conservation laws with AutoReduce are mass conservation laws, such as the total RNA polymerase being conserved as the sum of free polymerase, and all species in the model with a bound polymerase. Similarly, we have conservation laws for total DNA, total ribosomes, total endonucleases, and other resources.

After applying conservation laws, we used quasi-steady state approximation (QSSA) to further reduce the models. The built-in automated QSSA tools in AutoReduce were used for this purpose. By applying QSSA iteratively to reduced models, we obtain further reduced models. The details of the species that were assumed to be at QSS are shown in Figures 3.4 and 3.8. For both full models (the integrase action and integrase-excisionase action), we obtain a set of reduced models, each with different assumptions in the derivation of the reduced model. These are shown as flowcharts in Figures 3.4 and 3.8. Some reduced models were amenable to even further reduction in dimensionality by introducing species abundance assumptions. For example, on assuming abundance of RNA polymerase (P_{tot}), we may write the terms like $P_{\text{tot}} - C$ as approximately equal to P_{tot} where C is an intermediate low concentration complex species. Such assumptions were only true under certain parameter conditions and for certain reduced models, hence, did not always lead to correct reduced models (see the section on performance metrics in Methods for more on how we validate a reduced-order model). Nevertheless, with the Python tools in AutoReduce and our further additions to it, exploration of the space of models is quick. Finally, we extended AutoReduce to develop wrappers for easy import and export of SBML files so that the tool can be seamlessly connected to existing pipelines.

Context abstraction with BioCRNpyler

BioCRNpyler allows the modeling of biological systems in different contexts and modeling details with the use of built-in `Mixtures` (objects that model the context-dependent and global effects) and `Mechanisms` (objects that capture the modeling detail for a process). In this chapter, we utilize this key functionality of BioCRNpyler to explore the design space and possible hypothesis for integrase and excisionase-based DNA recombination. We developed two integrase action mechanisms — a simple mechanism that models the flipping of attP-attB promoter sequence to attL-attR in one step, and a detailed mechanism that models the same process but with binding events involving integrase binding to the different DNA sites. For excisionase action, the exact mechanism and binding steps are unknown, so we model all possible steps and explore the hypothesis with our iterative pipeline and the experimental data. We model the excisionase binding to DNA already bound with integrase on the attP-attB or the attL-attR sites as the first hypothesis of the excisionase mechanism. Once excisionase binds at the attL-attR site already bound with integrase, it flips the sequence to attP-attB. Another potential mechanism by which excisionase may reverse the directionality of the promoter is by binding to the integrase first and forming an integrase-excisionase complex. This complex, when bound with the DNA at the attL-attR site, flips the promoter to attP-attB. Further, the sequestration mechanism for excisionase is also modeled by including the reactions in which the excisionase binds to the attP-attB site, hence, preventing integrase to act on it. We add all of these mechanisms to the library of BioCRNpyler mechanisms so that they can be used to build CRN models in different mixtures.

We use the `TxTlExtract` mixture in BioCRNpyler along with the mechanisms described above to build detailed CRN models. In the iterations of the pipeline, as described in Figure 3.4 and 3.8, we switch mechanisms `ON` or `OFF` to abstract the details of the model and switch `Mixtures` to build simpler or detailed models. In conclusion, the context abstraction with BioCRNpyler is achieved by building mechanisms and mixtures and simply choosing the mixture to use for context and the mechanisms for the modeling detail.

Reduced model performance metrics

Each reduced model obtained in this chapter was tested for its performance against the full model by computing different metrics. We used the metrics developed in Chapter 2 of this thesis for this purpose. First, we computed the normed error metric, ζ, for each reduced model:

$$\zeta = ||y - \hat{y}||_2^2 ,$$

where y and $\hat{y}$ are the outputs of the full and the reduced model, respectively. In all cases analyzed in this chapter, we derived reduced models where the outputs of the full model were never collapsed since these are the signals which are measured. For most cases, the error performance metric sufficed in rejecting the reduced models, however, two other metrics that were shown to be useful in deciding among equally good error-performance models are:

1. Error sensitivity: It was shown that when normed error performance is of the same order for two or more reduced models, we can compute the sensitivity of the error with respect to each parameter in the model to choose a reduced model whose error is least sensitive to parameter perturbations.
2. Input-output map: For each reduced model, we computed an input-output system gain using linear systems theory [69] by first linearizing the model around a point of interest. This input-output gain was then compared with the gain of the full model to ensure the fidelity of the reduced models.

Using these error metrics, we decided which reduced model to choose as shown in Figures 3.4 and 3.8. The computation of the metrics for all reduced models shown in this chapter is available with the associated Python code on Github [124].

Data analysis and parameter inference

Standard Python libraries (NumPy [127] and Matplotlib [128]) were used for the analysis and plotting of experimental data. Three main analysis and optimization tools were used in this chapter:

Numerical simulations

The Cython-based tool Bioscrape [31] provides access to fast simulators suited to simulate CRN models even under stiff conditions that are commonly observed in mass-action ODE simulations. Although Bioscrape provides both deterministic and stochastic simulation tools we used only deterministic simulations for the analyses in this chapter. The Python wrappers available in Bioscrape were used to import SBML files from BioCRNpyler and AutoReduce to run simulations. All simulations shown in this chapter were performed using Bioscrape.

Sensitivity analysis

We extended the suite of analysis tools in Bioscrape by developing local sensitivity analysis methods. The local sensitivity coefficients are computed at each time point as

$$s_{ij} = \frac{dx_i}{dp_j},$$

where x_i is the i-th state and p_j is the j-th parameter. The sensitivity coefficients are arranged in a tensor of size length of timepoints × number of parameters × number of states. The computation of sensitivity coefficients is done by using the direct method [80]. Various options to control the accuracy, normalization, and integration methods are available to the user. Full documentation of the sensitivity analysis was added to the Bioscrape documentation on Github [129]. In Figures 3.3 and 3.7, we used the sensitivity analysis method to assess which parameters were most effective for each model. In Figures 3.5 and 3.9, we used the sensitivity analysis to find identifiable parameters from the data and guide a by-parts parameter inference process.

Bayesian inference using MCMC

We used time-series fluorescence reporter measurements to validate the mathematical models. Towards that end, we optimized the parameters in the model using Bayesian inference. We find probability distributions for parameters given the experimental data (posterior distributions) and then sample the parameter values from the posterior to simulate the model to plot alongside the data. Bayes' rule is the underlying principle for this approach:

$$\begin{aligned}\mathbb{P}(p_j|y_i) &= \frac{\mathbb{P}(y_i|p_j) \times \mathbb{P}(p_j)}{\mathbb{P}(p_j, y_i)} \propto \mathbb{P}(y_i|p_j) \times \mathbb{P}(p_j)\\ \log\left(\mathbb{P}(p_j|y_i)\right) &\propto \log \mathbb{P}(y_i|p_j) + \log \mathbb{P}(p_j),\end{aligned}$$

where p_j consists of the parameters for which we want the probability distribution given the data instance(s) y_i. The probability distribution $\mathbb{P}(y_i|p_j)$ is called the model likelihood since it is proportional to the likelihood of seeing the data y_i given that the parameters take the value p_j. We implemented multiple ways in which this likelihood can be computed by simulating the model at the given parameter values p_j and comparing the model against the data. For example, the likelihood may be computed with Bioscrape as the normed difference between the model outputs and experimental measurements over time or it may be computed as the maximum error between the simulation and the data. We implemented a total of 6 methods to compute the model likelihood. The Bioscrape documentation [129] describes these in detail.

For the parameter inference in this chapter, we used the deterministic trajectory likelihood which computes the normed error between the model and the data for all time in the data trajectories. The probability distribution $\mathbb{P}(p_j)$ is the prior probability distribution that gives each parameter value a probability of being the true value from the prior information about the parameters. We implemented multiple prior probability distributions in Bioscrape including, Gaussian, uniform, log-normal, log-uniform, beta, and more. A custom function may also be used as a prior probability distribution for a parameter. Bioscrape documentation describes their usage in detail. Since all models used in this chapter are models that describe biophysical processes, the parameters have mechanistic meanings, hence, priors were used to constrain the parameter inference according to the biological prior knowledge about each parameter. To compute each parameter sample, we use the Python emcee [113] package that implements a Markov Chain Monte Carlo (MCMC) sampler.

This MCMC sampler proposes the next parameter sample by assessing how the previous sample performed. For more details on the sampling algorithm, the reader is referred to the emcee documentation [113]. In conclusion, we developed Bioscrape inference as a black-box wrapper that imports experimental data and an SBML model to be used for parameter inference with emcee. Code for all data analysis, parameter inference, and related documentation are available on Github [124].

APPENDIX A

A.1 Raw data and concentration calibration

The raw fluorescence data for the integrase and reporter circuit shown in Figure 3.2 is shown in Figure A1. For the excisionase expression (mScarlet fluorescence) and its action on the reporter, the raw fluorescence data for CFP (integrase expression), mScarlet (excisionase expression), and YFP (reporter) is shown in Figure A2. We calibrated the plate reader fluorescence reading to obtain absolute concentration measurements for the cell-free experiments. The calibration factors are given in Table A1 where arbitrary units [AU] = Slope · [uM] + Offset.

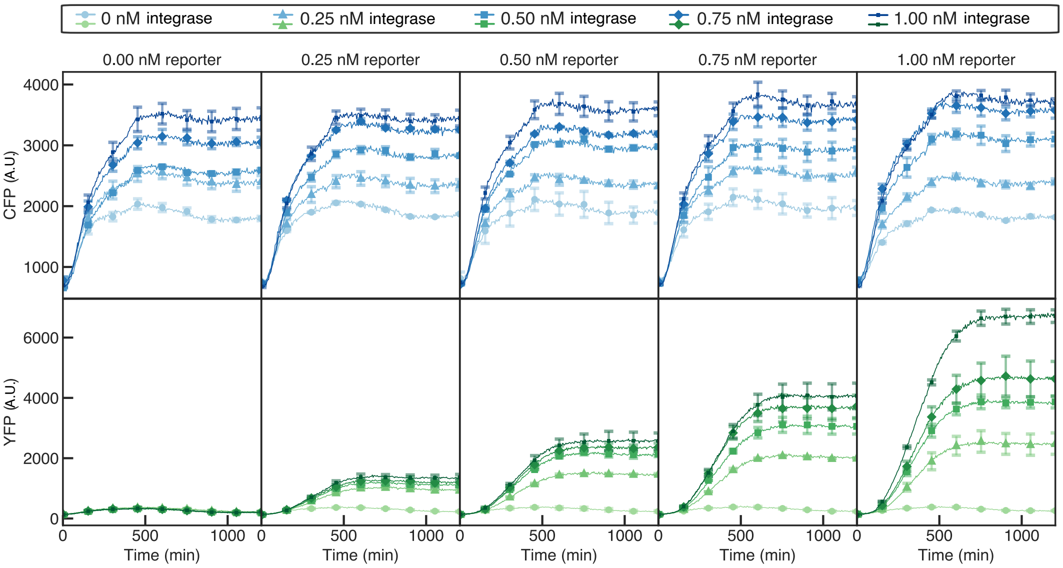


Figure A1: Cell-free fluorescence data for the integrase and reporter expression circuit. The raw measurements are in arbitrary units. We process these measurements by subtracting the background and calibrating the fluorescence to concentration units. Note that error bars are only shown once in every 30 points for clarity.

Measurement	Slope	Offset
CFP	12796.23	-6248.64
mScarlet	9886.58	166.59
YFP	44451.55	294

Table A1: Calibration factors for cell-free experiments

A.2 Model species

In the integrase model (shown in Figure 3.2), each variable represents the concentration of the model species. G_i is the integrase plasmid, P is RNA polymerase

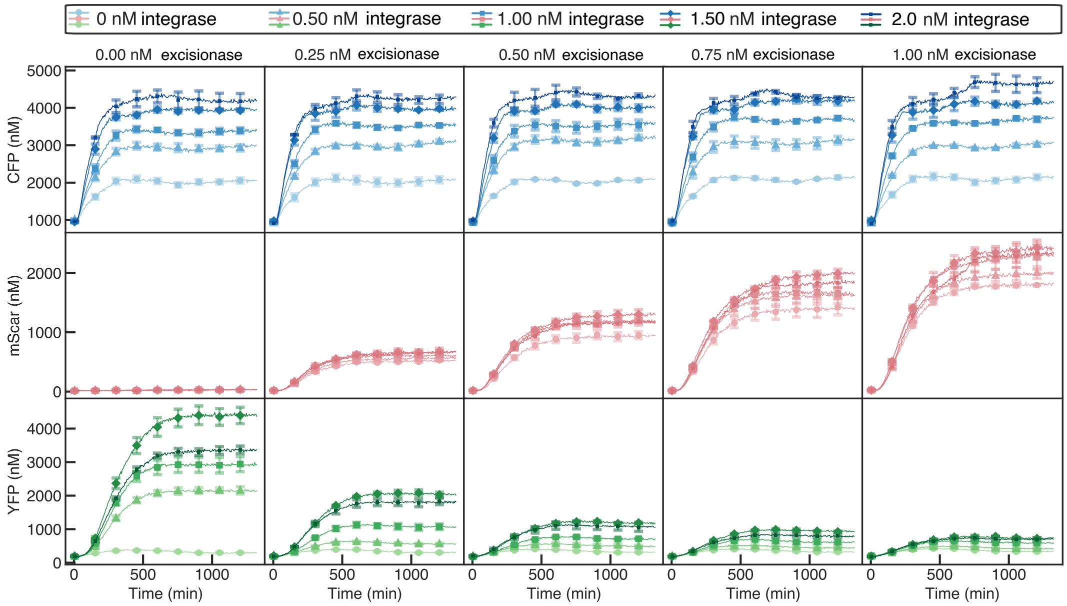


Figure A2: Cell-free fluorescence data for the integrase, excisionase, and reporter expression circuit. The raw measurements are in arbitrary units. We process these measurements by subtracting the background and calibrating the fluorescence to concentration units. Note that error bars are only shown for every 18 points.

(RNAP) and C_{i_1} is the binding complex between RNAP and the gene. T_i is the mRNA transcript that codes for the Bxb1-CFP (I) protein. The binding complex between mRNA and ribosome, R, is denoted as C_{i_2}. The reporter plasmid is called U in its “inactive” state (that is, when the promoter is in the reverse direction) and A in its “active” state. RNAP binds to the reporter gene to form the complex C_1 to transcribe the reporter mRNA T, which then binds to the ribosome (R) to form C_2. The YFP concentration is denoted as Y. For the integrase activity, we use n_i for the cooperativity of integrase binding to the inactive gene, U, to form the complex, C_i. The integrase flipping reaction forms the active complex, C_a, which then reversibly forms the active reporter plasmid, A.

We model the degradation of mRNA and its complexes by the endonucleases in cell-free extracts. The endonuclease concentration is denoted by E, which forms complexes C_{e_i}, C_{er_i}, C_e, and C_{er} when it binds to the integrase mRNA transcript, the ribosome-bound integrase mRNA transcript, the reporter transcript, and the ribosome-bound reporter mRNA transcript, respectively.

In the integrase-excisionase model (shown in Figure 3.6), we have the transcription and translation of the Xis-mScar gene (G_e) that expresses the fused excisionase-mScarlet protein (E_0). The corresponding transcript is denoted as T_e, the RNAP-gene complex as C_{e_1}, and the transcript-ribosome binding complex as C_{e_2}. For the excisionase action, we introduce two mechanisms. In the first mechanism, the excisionase, with a cooperativity of n_e, binds to the integrase-bound "active" reporter plasmid, C_a, with attL-attR sites to form the complex C_e. The excisionase flips the promoter direction in this complex to form C_r, which is the "inactive" complex with both integrase and excisionase bound at the attB-attP site. Then, this complex reversibly unbinds to give out free integrase, excisionase, and the "inactive" reporter plasmid, U. In the second excisionase mechanism, the excisionase binds to the integrase forming the complex C_{ie}. The excisionase may also bind to the integrase-bound DNA, C_i, forming C_r, to sequester the integrase from flipping C_i to C_a. Finally, the integrase-excisionase complex, C_{ie}, binds to both the "inactive" and the "active" DNA (U and A) to form complexes C_r and C_e, respectively. Similar to the endonuclease-mediated degradation of integrase and reporter mRNA and their complexes, the excisionase mRNA and its complexes are also degraded. The endonuclease binds to the excisionase mRNA to form C_{e_e} and it binds to the ribosome-bound excisionase mRNA, C_{e_2}, to form the complex C_{er_e}.

A.3 Parameter values

All parameter values and simulation conditions for all models in this chapter are available on Github [124]. To summarize some of these findings, we provide key parameter values for the detailed and minimal models as tables below. Note that we identified the posterior distributions (and hence, the uncertainties) for the parameters in the minimal models. The corner plots in Figures 3.5 and 3.9 show these uncertainties in each identified parameters. The reader is referred to the source code [124] for more details on statistical convergence, prior distributions, initial guesses, and posterior distributions with uncertainties.

Table A2: Cell-free global resources

Parameter	Description	Value	Unit
P_{tot}	Total RNAP in cell-free mixture	200	nM
R_{tot}	Total ribosomes in cell-free mixture	400	nM
E_{tot}	Total endonucleases in cell-free mixture	30	nM

Table A3: Integrase and reporter model parameters in cell-free extract

Name	Description	Value	Unit
a_i	Binding rate of integrase to attP-attB DNA site	10	nM$^{-n_i}$hour^{-1}
d_i	Unbinding rate of integrase from the DNA	1	hour^{-1}
k_{int}	Flipping rate of integrase	0.1	hour^{-1}
a_r	Unbinding rate of integrase and attL-attR complex	0.033	hour^{-1}
d_r	Binding rate of integrase and attL-attR site on DNA	0.5	nM$^{-n_i}$hour^{-1}
k_{bp_i}	Binding rate of RNAP to transcribe integrase-mRNA	100	nM^{-1}hour^{-1}
k_{up_i}	Unbinding rate of RNAP from the integrase DNA	1	hour^{-1}
k_{tx_i}	Transcription rate of integrase expressing mRNA	0.25	hour^{-1}
k_{br_i}	Binding rate of ribosome to the integrase mRNA	100	nM^{-1}hour^{-1}
k_{ur_i}	Unbinding rate of ribosome from the integrase mRNA	1	hour^{-1}
k_{tl_i}	Translation rate of integrase-CFP expression	0.013	hour^{-1}
n_i	Cooperativity of integrase action	3	–
k_{int}	Integrase flipping rate	0.1	hour^{-1}
k_{bp}	Binding rate of RNAP to transcribe reporter mRNA	80	nM^{-1}hour^{-1}
k_{up}	Unbinding rate of RNAP from the reporter DNA	2	hour^{-1}
k_{tx}	Transcription rate of the reporter mRNA	0.3	hour^{-1}
k_{br}	Binding rate of ribosome to the reporter mRNA	80	nM^{-1}hour^{-1}
k_{ur}	Unbinding rate of ribosome from the reporter mRNA	2	hour^{-1}
k_{tl}	Translation rate of the reporter	0.03	hour^{-1}
d_{bT_i}	Binding rate of RNAase to the integrase mRNA	20	nM^{-1}hour^{-1}
d_{uT_i}	Unbinding rate of RNAase from the integrase mRNA	2	hour^{-1}
d_{T_i}	RNAase mediated degradation rate of integrase mRNA	0.005	hour^{-1}
d_{bRT_i}	Binding rate of RNAase to the ribosome bound integrase mRNA	20	nM^{-1}hour^{-1}
d_{uRT_i}	Unbinding rate of RNAase from ribosome bound integrase mRNA	2	hour^{-1}
d_{RT_i}	RNAase mediated degradation rate of ribosome bound integrase mRNA complex	0.005	hour^{-1}
d_{bT}	Binding rate of RNAase to the reporter mRNA	10	nM^{-1}hour^{-1}
d_{uT}	Unbinding rate of RNAase from the reporter mRNA	2	hour^{-1}
d_T	RNAase mediated degradation rate of reporter mRNA	0.001	hour^{-1}
d_{bRT}	Binding rate of RNAase to the ribosome bound reporter mRNA	10	nM^{-1}hour^{-1}
d_{uRT}	Unbinding rate of RNAase from ribosome bound reporter mRNA	2	hour^{-1}
d_{RT}	RNAase mediated degradation rate of ribosome bound reporter mRNA complex	0.005	hour^{-1}

Table A4: Integrase-excisionase model parameters in cell-free extract

Name	Description	Value	Unit
a_{e_1}	Binding rate of excisionase to integrase bound attL-attR site on the DNA	2	nM$^{-n_e}$hour^{-1}
d_{e_1}	Unbinding rate of excisionase from integrase bound attL-attR site on the DNA	0.01	hour^{-1}
a_{s_1}	Binding rate of integrase to excisionase	0.02	nM^{-1}hour^{-1}
d_{s_1}	Unbinding rate of integrase from excisionase	1	hour^{-1}
a_{e_2}	Binding rate of integrase-excisionase complex to the attL-attR site on the DNA	1	nM$^{-n_e}$hour^{-1}
d_{e_2}	Unbinding rate of integrase-excisionase complex from the attL-attR site on the DNA	0.02	hour^{-1}
a_{s_2}	Binding rate of excisionase to integrase bound attP-attB site on the DNA	0.02	nM$^{-n_e}$hour^{-1}
d_{s_2}	Unbinding rate of excisionase from integrase bound attP-attB site on the DNA	0.05	hour^{-1}
a_{r_1}	Unbinding rate of integrase, excisionase, and the attP-attB site on the DNA from their complex	0.02	hour^{-1}
d_{r_1}	Binding rate of integrase, excisionase, and the attP-attB site on the DNA to form a complex	0.01	nM$^{-(n_e+n_i)}$hour^{-1}
a_{r_2}	Binding rate of integrase-excisionase complex to the attP-attB site on the DNA	0.02	nM$^{-n_e}$hour^{-1}
d_{r_2}	Unbinding rate of integrase-excisionase complex and the attP-attB site on the DNA from their complex	0.01	hour^{-1}
k_{bp_e}	Binding rate of RNAP to transcribe excisionase mRNA	100	nM^{-1}hour^{-1}
k_{up_e}	Unbinding rate of RNAP from the excisionase DNA	1	hour^{-1}
k_{tx_e}	Transcription rate of excisionase expressing mRNA	0.292	hour^{-1}
k_{br_e}	Binding rate of ribosome to the excisionase mRNA	100	nM^{-1}hour^{-1}
k_{ur_e}	Unbinding rate of ribosome from the excisionase mRNA	1	hour^{-1}
k_{tl_e}	Translation rate of excisionase-mScarlet expression	0.011	hour^{-1}
n_e	Cooperativity of excisionase action	4	–
k_{exc}	Excision rate	0.7	hour^{-1}
k_{tx}	Transcription rate of the reporter-mRNA	0.4	hour^{-1}
k_{tl}	Translation rate of the reporter	0.08	hour^{-1}
d_{bT_e}	Binding rate of RNAase to the excisionase mRNA	20	nM^{-1}hour^{-1}
d_{uT_e}	Unbinding rate of RNAase from the excisionase mRNA	2	hour^{-1}
d_{T_e}	RNAase mediated degradation of excisionase mRNA expression	0.005	hour^{-1}
d_{bRT_e}	Binding rate of RNAase to the ribosome bound excisionase mRNA	20	nM^{-1}hour^{-1}
d_{uRT_e}	Unbinding rate of RNAase from ribosome bound excisionase mRNA	2	hour^{-1}
d_{RT_e}	RNAase mediated degradation rate of ribosome bound excisionase mRNA complex	0.005	hour^{-1}

Note that some integrase parameters are updated in the parameter identification process as the context changes in this more complex circuit.

A.4 Parameter inference and unidentifiability

MCMC sampler for inference of integrase-reporter circuit

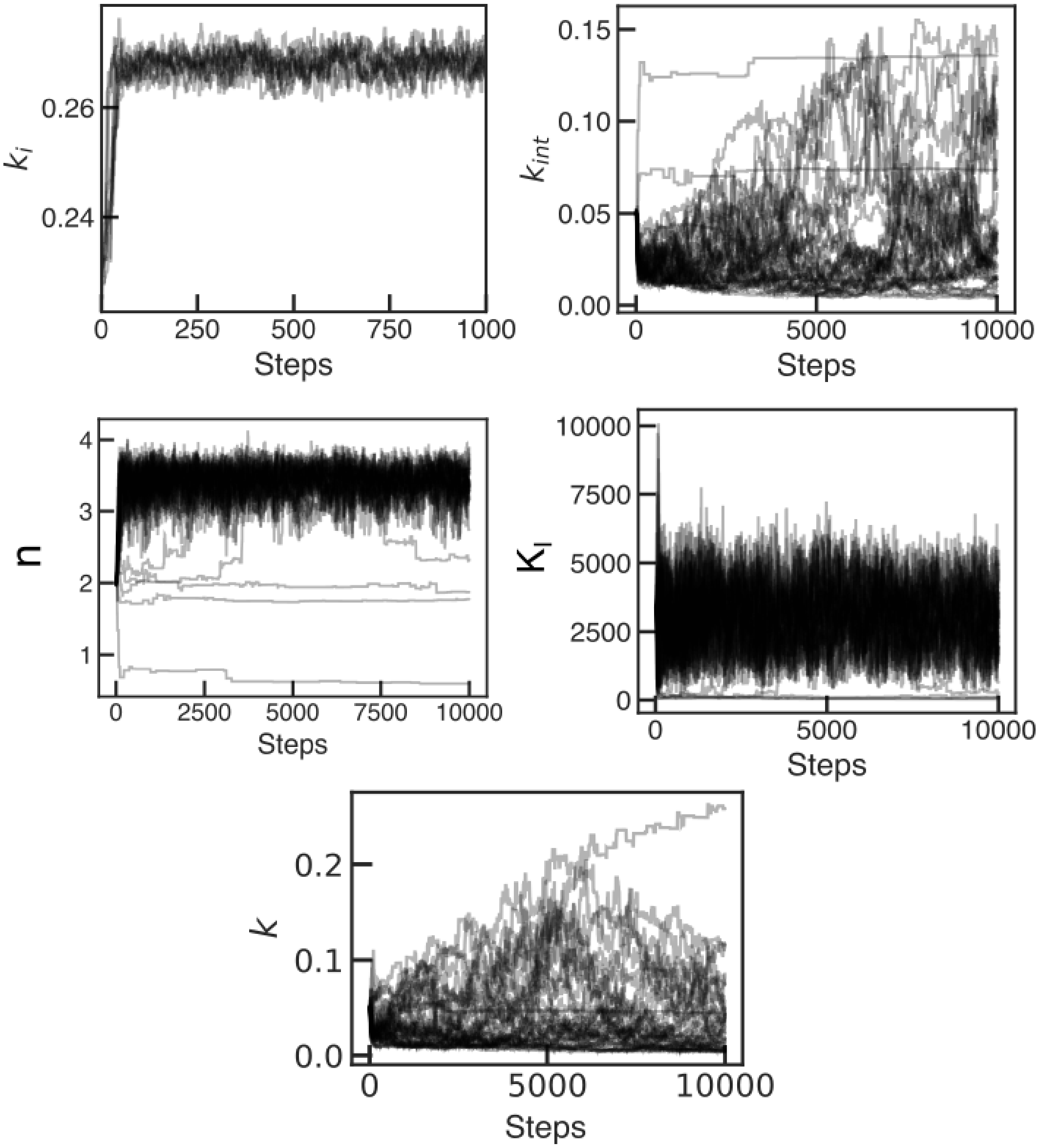


Figure A3: MCMC chains for identification of integrase model parameters given the integrase-reporter cell-free experiments.

In this section, we show the parameter inference results in detail for the integrase model and the integrase-excisionase model. The parameter inference uses experimental data from cell-free experiments. The same cell-free mixture (extract and buffer) is used for all experiments shown in this chapter to ensure that the context-dependent parameters remain constant throughout. For the integrase expression and action (model shown in Figure 3.2), the MCMC chains for the identified parameters are shown in Figure A3.

As discussed in the main text, we use system identification by parts to identify parameters sequentially using the sensitivities of the measurements to different parameters. The first step of parameter identification uses CFP measurement to identify k_i as suggested by sensitivity analysis (see Figure 3.5B). We run the MCMC sampler for 1000 steps, 10 walkers, and a Gaussian prior on k_i with a mean value of 0.23 and a standard deviation equal to 1. To identify the integrase action and reporter expression parameters, k_{int}, n, K_I, and k, we use YFP measurement as suggested by the sensitivity analysis. For this MCMC sampler, we use 40 walkers for 10000 steps and Gaussian priors on all parameters. The mean values used in the priors for k_{int}, n, K_I, and k are 0.05, 2, 3330, and 0.0001 while the standard deviations used are 10, 2, 1000, and 0.1, respectively. The total runtime for this parameter inference was 50 minutes on a personal computer with an Intel i7-6700 processor and 16GB of RAM.

Identified integrase model and data

We sample from posterior parameter distributions and run model simulations. We plot the simulations of the identified models together with the experimental data. These results are shown for both the CFP and YFP measurements in Figures A4 and A5, respectively.

MCMC sampler for inference of integrase-excisionase circuit

Here we describe the MCMC sampler used to infer the parameters of the integrase-excisionase and reporter circuit (model shown in Figure 3.6). The MCMC chains for the identified parameters are shown in Figure A6. First, we use the CFP measurement to re-estimate k_i in this updated context and then use the mScarlet measurement to infer k_e. See sensitivity analyses of the different outputs (CFP, mScarlet, and YFP) to model parameters in Figure 3.7. We run the MCMC sampler for 1000 steps, 10 walkers, and Gaussian priors on k_i and k_e with mean values set at 0.26 (previously identified maximum likelihood value) for k_i and 0.3 for k_e. Both priors are used with a standard deviation equal to 5. Further, to identify the integrase and excisionase action, and reporter expression parameters, k_{int}, d_i, a_{e_2}, a_{s_1}, d_{s_1}, and k, we use YFP measurements as suggested by the sensitivity analysis. For this MCMC sampler, we use 20 walkers for 20000 steps and Gaussian priors on all parameters. The mean values used in the priors for k_{int}, d_i, a_{e_2}, k_e, a_{s_1}, d_{s_1}, and k are 0.2, 500, 0.02, k_e (previously identified), 0.1, 100, and 2 while the standard deviations used are 5,

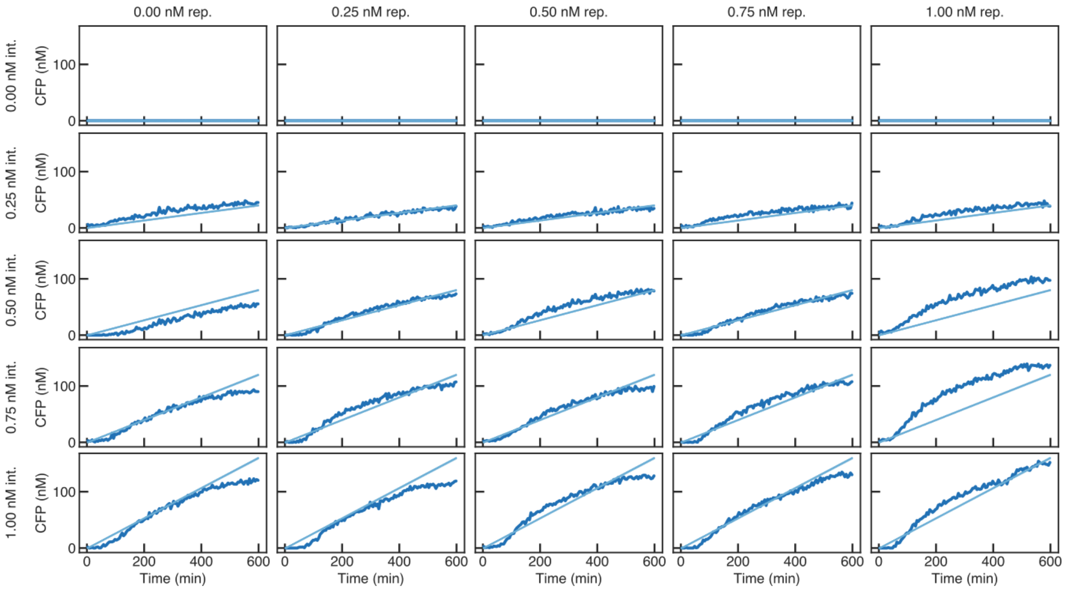


Figure A4: Model simulations with posterior parameter values plotted alongside CFP measurement. Observe that the model fits the data well for most conditions, however, for higher reporter plasmid concentration the fit worsens. See the discussion on loading effects in Material and Methods for more information on this observation.

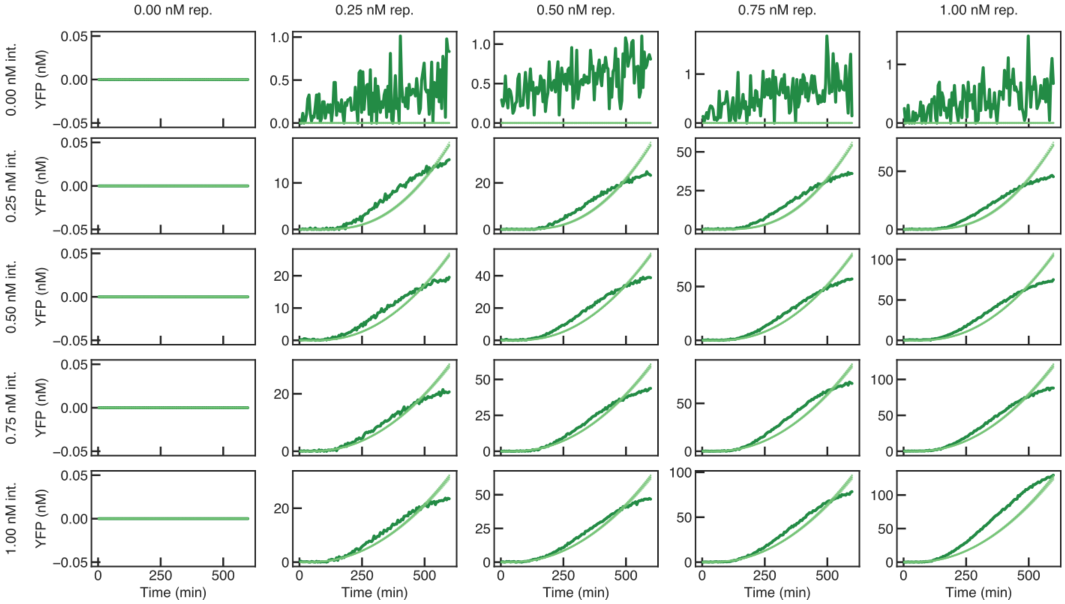


Figure A5: Model simulations with posterior parameter values plotted alongside YFP measurement. Observe that the minimal model (derived automatically from a detailed CRN model) does not fit the dynamics as cell-free extract stops protein expression. This is a result of context abstraction steps in obtaining this minimal model. Since the effects of resource usage are not modeled in this minimal model, the YFP expression in the model does not saturate as quickly as the observed data.

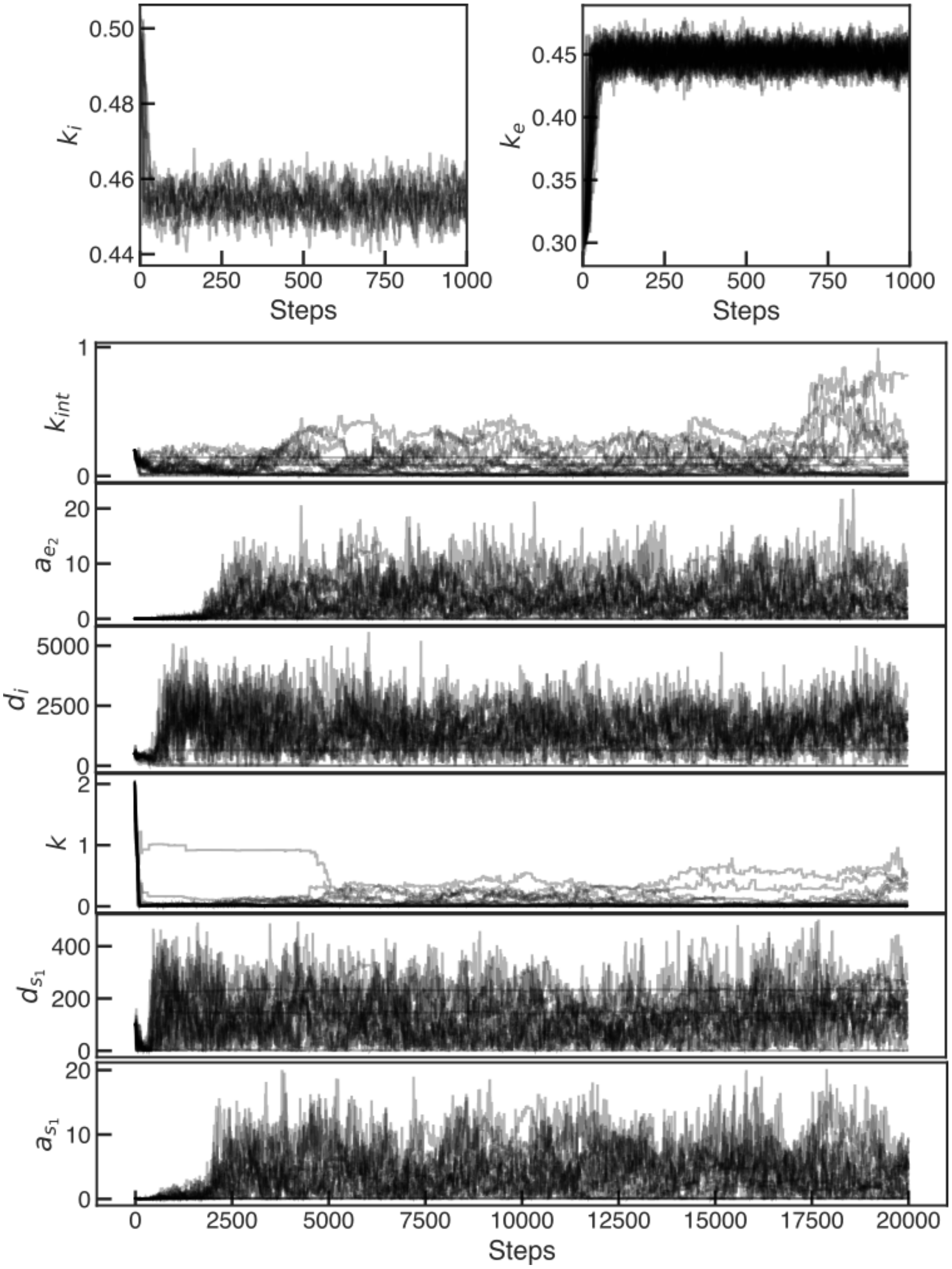


Figure A6: MCMC chains for identification of integrase-excisionase model parameters using the data from cell-free experiments.

1000, 5, $0.1 \cdot k_e$, 5, 100, and 10, respectively. The total runtime for this parameter inference was 24 hours on a personal computer with an Intel i7-6700 processor and 16GB of RAM.

Identified excisionase model and data

We sample from posterior parameter distributions and run model simulations. We plot the simulations of the identified models together with the experimental data. These results are shown for all three measurements: CFP, mScarlet, and YFP in Figures A7, A8, and A9, respectively. As it is clear from the runtime, an increase in the dimension of the parameter inference problem from 3 to 6 led to a significant increase in runtime from around 1 hour to 24 hours. This is an expected curse of dimensionality, which leads to difficulty in inferring the parameters of detailed models. In the model-data fits shown above, the CFP and mScarlet measurements agree with the fitted model simulations, but, the YFP measurements do not perfectly

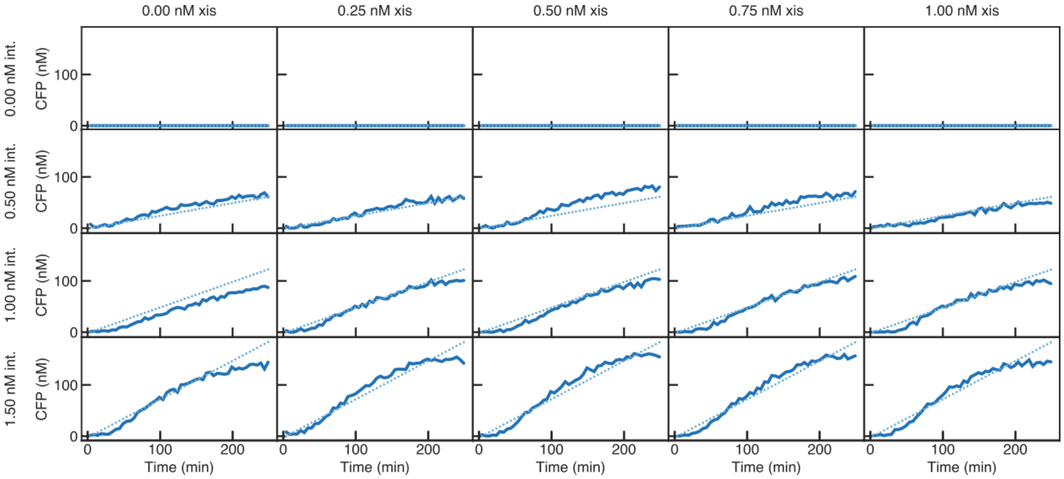


Figure A7: Model simulations with posterior parameter values plotted alongside CFP measurement.

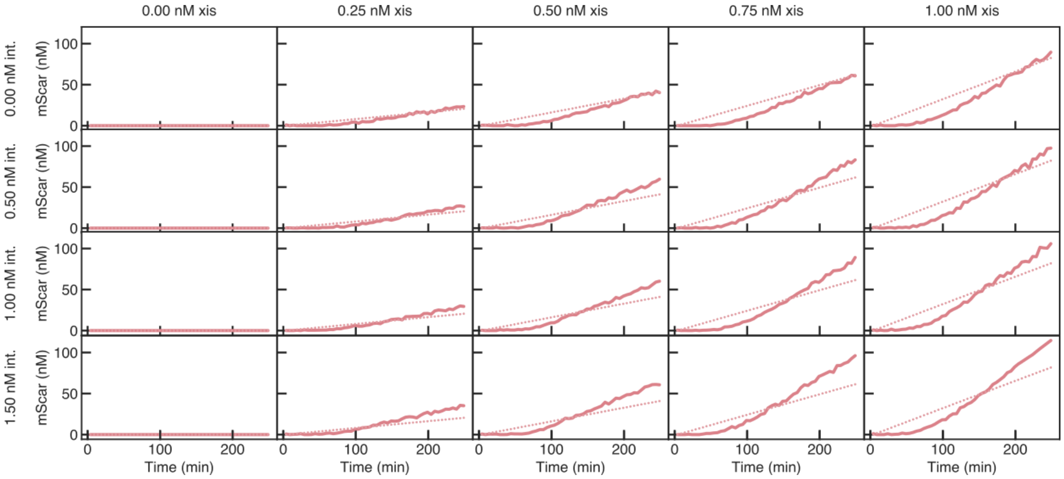


Figure A8: Model simulations with posterior parameter values plotted alongside mScarlet measurement.

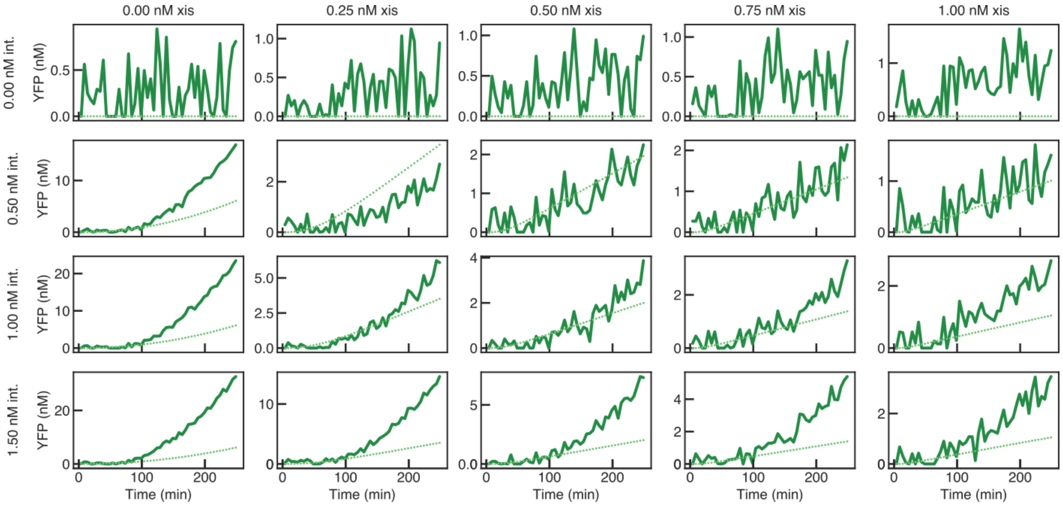


Figure A9: Model simulations with posterior parameter values plotted alongside YFP measurement. Note that although the model qualitatively captures the YFP expression but does not fit the data very well for some of the conditions. This is a result of a computational tradeoff in choosing a lower dimensional model for feasible parameter identification. The detailed model would capture the experimental behavior but it is infeasible to estimate 54 parameters with Bayesian inference.

fit the model predictions. The qualitative trends of the YFP expression are predicted correctly but since we are only identifying 6 parameters out of a total of 54 in the detailed model, these inaccuracies are expected.

Chapter 4

CONTRACT-BASED MODELING AND DESIGN

4.1 Main contributions

We provide a new perspective on using formal methods to model specifications and synthesize implementations for the design of biological circuits. In synthetic biology, design objectives are rarely described formally. We present an assume-guarantee contract framework to describe biological circuit design objectives as formal specifications. In our approach, these formal specifications are implemented by circuits modeled by ordinary differential equations, yielding a design framework that can be used to design complex synthetic biological circuits at scale.

We use a contract-based design software called Pacti [130] to aid the design of engineered biological systems. We consider a conditional guide RNA trigger-based [131] biological NAND logic gate with three subsystems. Two of the three subsystems can be chosen from a library of sensors that we have constructed using experimental data. We use the *quotient* operation on contracts to infer a specification of the missing third subsystem that needs to be designed. Further, once a nominal set of three subsystems have been chosen, we use contract *composition* as part of an optimization strategy to maximize the system fold-change, defined as the ratio between the on and off levels of the system's output.

Using the same approach of quotient-then-composition, we scale the design optimization to systems with five and then seven components. Finally, we show the design-space exploration of a large biological circuit with 1001 different possibilities to find successful designs.

4.2 Introduction

Research in synthetic biology has accelerated in the past decade due to its enormous promise in engineering biological systems for desired behavior. As discussed in Chapter 1, the synthetic biology applications range from biomedicine to engineered living materials and environmental remediation. Mathematical modeling has played a key role in the foundations of synthetic biology. In two papers published in 2000, dynamical systems were used to successfully analyze the oscillatory response of the repressilator [11] and the bistability in the toggle switch [10]. Since then, mathematical modeling has been extensively used to study the design of engineered biological systems [132]. Phenomenological models of gene regulation in synthetic biological circuits, such as activation and repression, have seen the most success in various applications [100], [133]–[135]. These models use simple nonlinear functions to model the biological circuit phenomena representing their input-output behavior. On the other hand, detailed models of mechanisms and metabolic pathways have long been used in systems biology to quantitatively study various properties of biological systems [136]. This kind of detailed modeling is important in engineered biological systems, as well. For example, chemical reaction network models of protease-based biosensors have been developed to engineer optimized logic gates [137]. Similarly, detailed mathematical models have been used to amplify biofuel production [138] and to characterize fluorescent protein maturation [139]. A review article on the current modeling practices in synthetic biology [140] provides further references on this topic.

The choice of using a detailed or a phenomenological model is made according to the problem at hand. From a design standpoint, the development of models is necessary to decide when to use the circuits described by these models. This is a central focus of this thesis as seen in Chapters 2 and 3. In Chapter 2, we demonstrated the importance of exploring the underlying assumptions in reduced models and proved conditions under which different gene expression models are valid. On similar lines, in Chapter 3, we explored reduced models of the integrase and excisionase enzymes in cell-free to predict the induction ratios. However, until now, all of these results have used the detailed modeling of the mechanisms of each component in the system. The curse of dimensionality is evident in the analysis shown in Chapter 3. The parameter inference-based system design for a system with 3 components took around 1 hour while for a system with 6 components took around 24 hours. As a result, the approaches discussed until this point in the thesis are not easily scaled to large system designs.

As system complexity increases, we believe that it is necessary to develop a complete design methodology that begins with a top-level description of the system's objective and guides the designer in the generation of an implementation that can be proved to meet the specification. This is the one of the main contributions of this chapter. Our methodology decouples reasoning about component specifications from reasoning about the modeling details of each component. This allows understanding the system's specification by analyzing the specifications of the subsystems. This methodology allows designers to focus on particular aspects of the design process at various levels of detail while ensuring that other aspects of the design are not forgotten. Current design approaches in synthetic biology do not model system specifications mathematically, but rather informally. As a consequence, the resulting mathematical models obtained are often disconnected from the specifications. It is thus difficult to answer whether the obtained models behave according to the specifications.

In this chapter, we apply the theoretical foundations for scalable engineered system design and related tools to enable compositional design of synthetic biology circuits. We use a recently developed contract-based design tool called Pacti [130], to certify system behavior against a specification through scales of model complexity. Using Pacti, we present a subsystem specification synthesis and composition strategy to predict system response. We validate these predictions with *in vivo* experimental data.

The state of the art of biocircuit design

The current synthetic biological circuit design process is largely heuristic-driven, where scientists makes design decisions based on their experience. Various modeling and computational tools assist the scientist in this process. Analysis of circuit design by developing ODE models and running simulations is used to validate proposed hypotheses. Other modeling formalisms such as chemical reaction-based or stochastic/partial differential equations are also used similarly. Design decisions and predictions are sometimes possible using such simulations. For example, in Chapter 3 of this thesis, different abstractions of chemical reaction network models are developed to predict the level of inducers to use in a system design with three components to achieve the desired output. Similarly, mathematical analysis of fixed points, oscillations [141], resource competition [142], and other modeling detail driven predictions are commonly employed in circuit design. But, a major disadvantage of this approach is that it does not scale well with the number of components in the circuit and hence cannot be used to predict larger circuit designs. Context-dependence of each component and modularity are big challenges that cannot be easily tackled in detailed modeling approaches.

Recently, we have seen a rise in computational design approaches that guide the engineering of proteins. For example, a design and optimization framework, called SAMPLE, [143] performs a combinatorial exploration of the DNA sequence space to maximize the protein fitness landscape. However, the protein design frameworks are limited to the design of proteins and not generalizable to bio-circuits where multiple proteins, enzymes, and other components interact to achieve complex system-level behavior.

For synthetic biological circuit design, design automation techniques and algorithms to propose circuit designs that achieve desired circuit behavior have been proposed. Cello [24] is a design tool that has been successfully used for designing large Boolean logic gate circuits based on a high-level language that describes the desired circuit behavior. It uses a library of characterized DNA parts for circuit design. On similar lines, a bio-design vision [144] was recently shared that vouches for co-design of hardware, software, and wetware together acting as compilers of large genetic circuits. Our design methodology shares similar principles where system designs are proposed in a modular way by using input-output relationships of each component. We discuss this in more detail next.

Scaling system designs

As discussed, mathematical models have been used to study the behavior of biological systems and analyze various properties. However, current experimental design approaches in synthetic biology are largely heuristics and trial-and-error based, hence, difficult to scale. We believe that the design framework presented here holds the potential to address this issue of scale due to its generality in writing system specifications and in decoupling the modeling details of each component from their specification. Using our approach, we can synthesize missing specifications given a larger system design and also predict behavior of systems by exploring the design space.

Our design methodology is centered on contract-based design [145]. At the heart of contract-based design are assume-guarantee contracts, which are formal specifications that distinguish between the responsibilities of a component and the assumptions made on its environment. Contracts come with a rich algebra. Contract-based methodologies have been developed for digital circuit design, aircraft power distribution systems [146], and certain classes of control systems [147]. Contract-based design is an engineering methodology used in systems engineering to provide modularity and guaranteed performance for systems with multiple components interacting in a complex fashion [148].

Recently, we developed a contract-based design and analysis software package called Pacti [130]. Pacti is an open-source Python package for carrying out compositional system analysis and design. Pacti represents components in a system using assume-guarantee specifications. Pacti can be used to obtain sensible system specifications from the specifications of the constituent subsystems. It can also be used to compute specifications of missing subsystems that need to be added to a design in order to meet an objective and diagnose inconsistent design errors when interconnecting components.

With the use of Pacti, we envision speeding up the experimental design process for synthetic biology in three ways: 1) by developing a characterized library of parts as assume-guarantee contracts using existing experimental data, 2) by constraining the design-space to automatically catch design errors and incompatible system components, and 3) by finding specifications of missing parts using the quotient operation on contracts.

We demonstrate the application of our methodology with the design of various engineered biological circuits — a transcriptional control-based AND logic gate, a dCas9-mediated NAND logic gate, and scaled *in silico* designs of biological circuits. We also validate our computational predictions with *in vivo* experimental data. We discuss how contracts help us to meet a system-wide specification when we have a partial implementation of the system available. Finally, we show how our methodology can be extended to seamlessly connect the specification of a component with its models and implementations.

From specifications to implementations

System design starts by writing the desired objectives mathematically so that any proposed implementation can be validated against the desired objectives. This practice of formally writing mathematical specifications is common in many areas of engineering. Often, it is possible to map these specifications to detailed implementation models [145], [149] as well to create system synthesis and integration pipelines. In this chapter, we explore an approach to write detailed models using domain knowledge for synthetic biological circuits starting from assume-guarantee contract specifications.

4.3 A contract-based design primer

Contract-based design [145] is a methodology by which a system is implemented by starting from a specification expressed as a contract and proceeding through a process of successive refinements, each of them adding increasing amounts of detail to the implementation.

The design process begins with a contract for the system we wish to implement and with a library of components that will be used to construct such a system. Each component in the library is represented by a contract. A mapping process identifies a set of elements from the library whose composition implements the top-level contract. If an implementation cannot be found, it may be the case that the library is insufficiently populated to implement the given specification. In this case, we can identify the contract specification of an element we need to add to the library to meet the specification. On the other hand, when an implementation is obtained by the mapping process, we have either reached our goal, or we may need to determine additional details of the candidate implementation. In the latter situation, the result of the mapping process can become the “top-level” contract input of the next mapping step that adds more details to the implementation. In other words, the design process can repeat as many times as needed until all details of the design are determined.

In contract-based design, the library of components consists of contracts; hence, we have to characterize as a contract every component we wish to add to our library. Since there are various kinds of analyses we may wish to perform on the system, there can be multiple contracts that can be associated with a given element. For example, we may keep separate the functionality specification from the performance. The multiple angles from which we can look at components are called *viewpoints*.

Operating on formal specifications offers the advantage that we keep separate the purpose of the system from its implementation details. This allows the designer to carry out analysis at the specification level, without getting distracted by detailed models. Moreover, by keeping separate the various viewpoints of the design, the designer can focus on a given type of analysis without worrying about the specifics of other aspects.

Formal aspects of assume-guarantee contracts

Assume-guarantee contracts are formal specifications that distinguish between (i) assumptions made on the environment and (ii) responsibilities attributed to the object being specified when it operates in an environment that meets the assumptions of the contract. An assume-guarantee contract C is thus a pair (a, g) of constraints denoting assumptions and guarantees, respectively. We say that a component is an environment for the contract if it meets the constraints a; we say a component is an implementation for a contract if it meets the guarantees g provided it operates in an environment for the contract, i.e., if it meets the constraint $a \to g$, where the arrow is logical implication (i.e., $a \to b = a \vee \neg b$). For a treatment of assume-guarantee contracts and their algebraic aspects, see [150]–[153].

We assume that constraints come from a Boolean algebra, i.e., there are well-defined notions of conjunction, disjunction, and negation for constraints. The Boolean algebra of constraints generates a partial order on contracts called refinement. We say that $C = (a, g)$ refines $C' = (a', g')$ (or that C' abstracts C) when the environments of C' are environments of C and the implementations of C are implementations of C', i.e., when $a' \leq a$ and $a \to g \leq a' \to g'$. We say that two contracts are equivalent if they have the same environments and the same implementations. Note that any contract (a, g) is equivalent to the contract $(a, a \to g)$. A contract in this form is said to be in saturated or canonical form.

Suppose we have two components obeying contracts C and C', respectively, and we want to obtain the specification of the system built using these two components. The contract operation of *composition* gives us the smallest contract in the refinement order obeyed by the system.

The closed-form expression for composition [152] is

$$C \parallel C' = \begin{pmatrix} (a \wedge a') \vee (a \wedge \neg g) \vee (a' \wedge \neg g'), \\ (a \to g) \wedge (a' \to g') \end{pmatrix}.$$

The composition operation is monotonic in the refinement order, i.e., composing by bigger contracts yields bigger results. As the composition is the smallest contract obeyed by the system, the system also obeys any abstraction of the composition operation. This observation will be used to provide results of composition that are closer to the intuition of a designer.

In addition to building systems using components, we are sometimes interested in finding components that allow us to meet a goal system-level specification. Suppose we wish to implement a system having a specification C, and we have available a subsystem obeying a contract C'. We want to find the specification of a second subsystem whose composition with the existing subsystem satisfies the top-level specification C. The largest such contract in the refinement order is given by the operation of *quotient*, whose explicit form [151] is

$$C/C' = \begin{pmatrix} a \wedge (a' \to g'), \\ (a' \wedge g) \vee \neg a \vee (a' \wedge \neg g') \end{pmatrix}.$$

Since the quotient is the biggest specification that completes the system, any contract smaller than the quotient also completes the system. As before, we will find it convenient to refine the quotient to provide contracts that match the intuition of the designer.

In addition to composing and decomposing systems, contracts offer support for multi-viewpoint design [154]. As we discussed, we are also interested in carrying out analysis of systems by focusing on one aspect of the system. This aspect may be functionality or performance. This means that we can assign to each component in our system several contracts, one for each viewpoint. The operation of *contract conjunction*, or *weak merging*, can be used to summarize into a single contract two viewpoints of the same object. Conjunction is given by the expression

$$C \wedge C' = (a \vee a', (a \to g) \wedge (a' \to g')).$$

4.4 Contract-based design in synthetic biology

We explore the application of Pacti to aid the design of engineered biological systems. First, we consider a NAND logic biological circuit design.

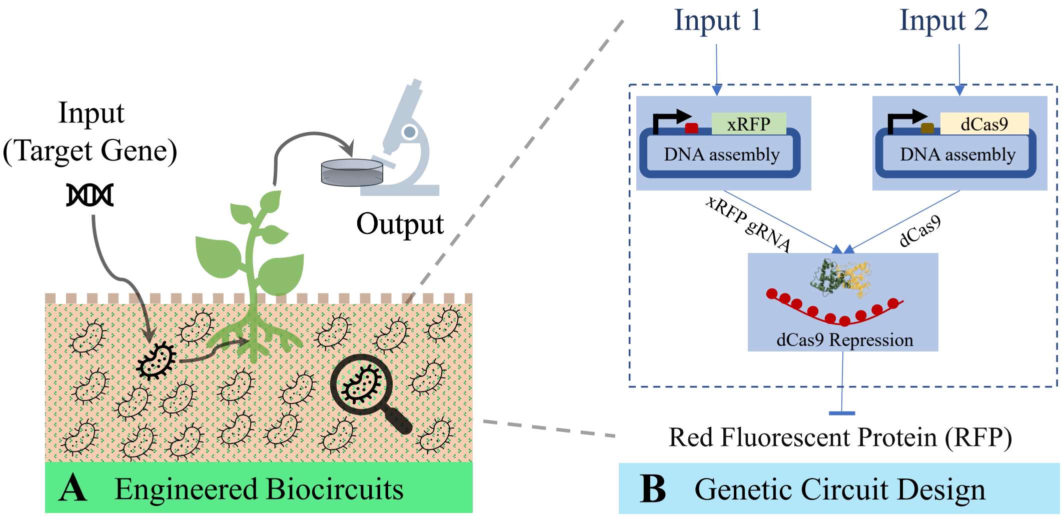


Figure 4.1: Specification-based synthetic biology using Pacti. (A) The goal with the biocircuit design is to measure the plant protein expression under control of engineered bacteria near its roots. (B) The bacterial cell is engineered as a NAND logic gate.

4.5 NAND gate design

Consider the system schematic shown in Figure 4.1a, where we are interested in measuring the level of a protein expressed by a plant under control of bacteria in the rhizosphere. The input in the overall system design is a target gene of interest introduced in the soil through bacteria. The goal is to measure the protein expression of the plant in response to the target gene input. The way we detect whether this gene is present is with many different complex biological processes and the output of whether we detect the gene, is read out from the plant leaf. So, the input is the gene of interest that we want to detect and the output is the protein production out of the plants. Designing such a biological system is difficult due to the lack of information about key biological mechanisms, absence of characterization data for system parts, unknown effects of the soil context, and many other context-dependent factors. There are many biological processes that would need to be engineered to build a system like this. In this chapter, we only focus on the design of a small part of the overall system — the engineered bacteria acting as a NAND logic biosensor.

This NAND gate system has two inputs and one output. To sense the two inputs, we need to design two sensing subsystems. A third subsystem that represses the top-level system output when both inputs are sensed is also required. In this way, a minimum of three subsystems are required to build a NAND logic system. A possible biological implementation of the NAND logic is shown in Figure 4.1b.

The two sensing subsystems can be chosen from a library of sensors that we have constructed using experimental data from the literature. We use the *quotient* operation on contracts to infer a specification of the missing third subsystem that needs to be designed. Further, once a nominal set of three subsystems have been chosen, we use contract *composition* as part of an optimization strategy to maximize the system fold-change, defined as the ratio between the on and off levels of the system's output. We say that an input or an output is "ON" when its level is higher than a minimum threshold, and is "OFF" when its level is lower than a maximum threshold. We formalize these notions with polyhedral constraints in the contract descriptions.

For the candidate NAND gate design, we have a sensor with tetracycline (aTc) input that outputs a dCas9 protein, a Salicylate (Sal) sensor that outputs the xRFP-guide RNA, and a dCas9 repressor subsystem, which takes as inputs the xRFP-guide RNA and the dCas9 proteins. When both inputs to this repressor subsystem are ON, it suppresses the output [131], RFP, a red fluorescent protein. RFP is also the output of the top-level system. RFP is ON only when either of the sensors inputs are OFF. In this way, the system behavior is that of a NAND logic gate. We will denote the contract for the sensors as $C_{\mathrm{aTc}}, C_{\mathrm{Sal}}$, and the contract for dCas9 repression mechanism as C_{dCas9}.

We address the following three design tasks in this chapter:

1. **Characterized library of parts as assume-guarantee contracts:** Using existing experimental data from the literature, we present a library of sensors as assume-guarantee contracts that can be used to reason about specifications of diverse biological circuits. In particular, we use the transcription factor-based sensor array in the "Marionette" cell lines [100] to construct a library of 14 sensors. In this way, we write the contracts C_{aTc} and C_{Sal} from the library of sensors.

2. **Finding specifications of missing parts:** In synthetic biology, it is common to have parts in the system for which no characterization data is available. Using quotient operation on contracts, we can find the constraints that this missing part must satisfy to meet the desired top-level criteria. Here, we find the missing contract of the dCas9-mediated repression mechanism, C_{dCas9}, for which characterization data is not easily measurable.

3. **Speeding up the experimental design:** Current experimental design approaches in synthetic biology are heavily reliant on screening experiments where all possible choices are explored experimentally to choose the one that works. This is not only time consuming and expensive but also does not scale beyond three or four independent choices. With the use of Pacti, we demonstrate how scientists may describe the desired top-level behavior as contracts and then computationally choose from a library of available parts to ensure that the components meet the top-level system specification. In this chapter, we find the specification of the sensors that meet the top-level criteria on fold-change of the circuit response. Given an implementation of C_{dCas9} and for a fixed choice of subsystem 2: C_{aTc}, we aim to find the best choice for the first sensor (choose specification for subsystem 1) that maximizes the top-level system fold-change.

The top-level system specifications

For the top-level system design, we write the contracts for a system with two inputs and one output such that it behaves like a NAND logic gate. We use an experimental implementation of a working NAND gate as a starting point to calibrate the desired contract specifications to real experimental measurements. For the NAND gate in Figure 4.1B, we have the experimental data for the implementation where input 1 is the Salicylate (Sal) sensor that is engineered to express the xRFP guide RNA and the input 2 is the tetracycline (aTc) sensor that expresses dCas9. The measured output is the fluorescence of the red fluorescent protein signal. The circuit diagram is shown in Figure 4.2a and the ideal NAND logic behavior is shown in Figure 4.2b.

The control in the circuit design measures the RFP signal with changing Sal concentrations but with a non-functional guide RNA. This is the leaky response of the circuit, and we use this data to compute the leak-related parameters in the contracts. We use the data for titration of the Sal input to characterize the linear regime contracts that model the transition of the input from zero to higher values and corresponding decrease in the RFP values from high (leaky) values to lower values.

Since the experimental data does not contain the titration for aTc, we synthetically create the data for how the output varies with aTc (mirroring the response of RFP to Sal). The *in vivo* experimental data for this NAND gate is shown in Figure 4.2c. This is a preliminary experimental data with one replicate for the

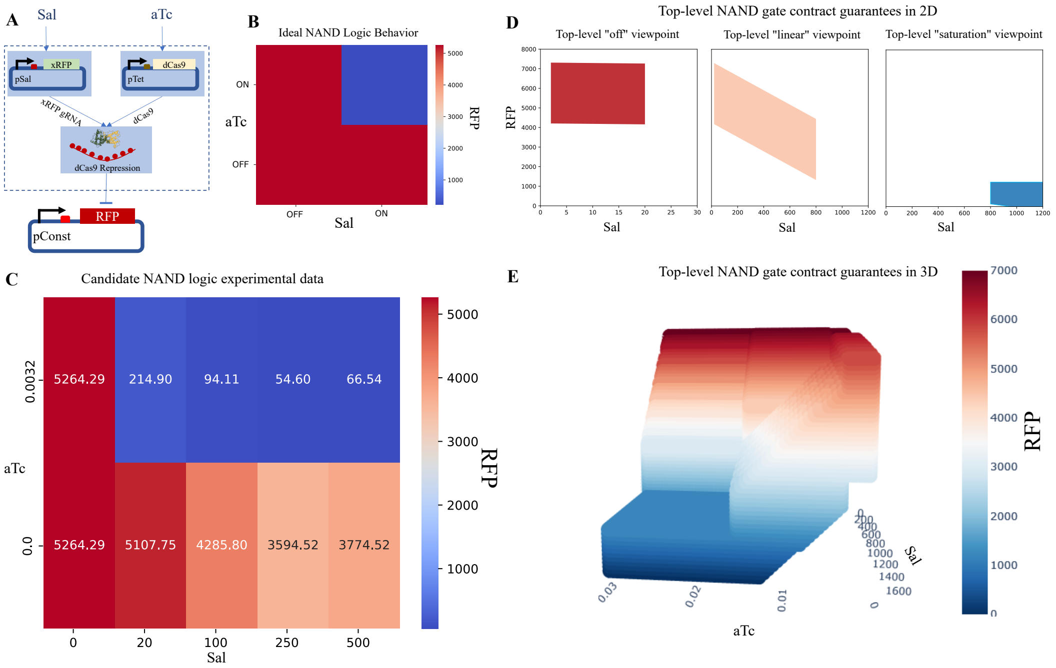


Figure 4.2: The desired top-level NAND logic contracts using a candidate circuit implementation. (A) The candidate NAND logic circuit implementation using the Sal and the aTc sensors for subsystem 1 and 2. (B) Ideal NAND logic behavior truth table. We desire a low output value (shown in blue) when both inputs are high and the output remains at a high value (shown in red) for all other conditions. (C) The experimental data for the candidate NAND logic circuit implementation (data courtesy of Shuwen (Eric) Lei, Caltech). (D) The top-level contract guarantees for different viewpoints – off, linear, and saturation. These are 2D plots for guarantees with aTc fixed at a high value and Sal being the independent variable changing on the X-axis. These correspond to the experimental setup. Finally, in (E) we show the 3D illustration of all NAND logic top-level system contract guarantees.

NAND gate implementation in an engineered Marionette strain of E. coli where RFP fluorescence was measured. We observe that as soon as Sal reaches a high value of 20 when aTc is also high, the RFP signal drastically reduces from 5264 to 214 (arbitrary units). However, when the guide RNA is not functional, we observe leaky RFP expression with changing values of Sal. The leaky output response is shown corresponding to the row labeled with aTc 0, which is a proxy for the dCas9 repression of RFP being non-functional. We create three viewpoints for the top-level contracts: the off region, the linear region, and the saturation region.

For each of the conditions of the NAND logic behavior, we write the contracts:

$$
\begin{aligned}
C_{\text{sys}}^{\text{off,off}} =&(0.000325 \leq \text{aTc} \leq 0.00325 \wedge 2 \leq \text{Sal} \leq 20,\\
&4211 - 2.566 \cdot \text{Sal} - 12.83 \cdot \text{aTc} \leq \text{RFP}\\
&\qquad\qquad \leq 7317 - 2.566 \cdot \text{Sal} - 12.83 \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{lin}_1} =&(0.00325 \leq \text{aTc} \leq 0.0156 \wedge 20 \leq \text{Sal} \leq 800,\\
&4251 - 5.121 \cdot \text{Sal} - 146.3 \cdot \text{aTc} \leq \text{RFP}\\
&\qquad\qquad \leq 7370 - 5.414 \cdot \text{Sal} - 146.3 \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{on,on}} =&(0.0156 \leq \text{aTc} \leq 0.0312 \wedge 800 \leq \text{Sal} \leq 1600, 873.6 - 0.9193 \cdot \text{Sal}\\
&- 0.1839 \cdot \text{aTc} \leq \text{RFP} \leq 1250 - 2.02 \times 10^{-5} \cdot \text{Sal} - 2.02 \times 10^{-5} \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{on,off}} =&(0.0156 \leq \text{aTc} \leq 0.0312 \wedge 2 \leq \text{Sal} \leq 20,\\
&4211 - 2.566 \cdot \text{Sal} - 0.1839 \cdot \text{aTc} \leq \text{RFP} \qquad\qquad (4.1)\\
&\qquad\qquad \leq 7317 - 2.566 \cdot \text{Sal} - 2.02 \times 10^{-5} \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{lin}_2} =&(0.0156 \leq \text{aTc} \leq 0.0312 \wedge 20 \leq \text{Sal} \leq 800,\\
&4251 - 5.121 \cdot \text{Sal} - 0.1839 \cdot \text{aTc} \leq \text{RFP}\\
&\qquad\qquad \leq 7370 - 5.414 \cdot \text{Sal} - 2.02 \times 10^{-5} \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{off,on}} =&(0.000325 \leq \text{aTc} \leq 0.00325 \wedge 800 \leq \text{Sal} \leq 1600,\\
&4211 - 0.9193 \cdot \text{Sal} - 12.83 \cdot \text{aTc} \leq \text{RFP}\\
&\qquad\qquad \leq 7317 - 2.02 \times 10^{-5} \cdot \text{Sal} - 12.83 \cdot \text{aTc}),\\
C_{\text{sys}}^{\text{lin}_3} =&(0.00325 \leq \text{aTc} \leq 0.0156 \wedge 800 \leq \text{Sal} \leq 1600,\\
&4251 - 0.9193 \cdot \text{Sal} - 146.3 \cdot \text{aTc} \leq \text{RFP}\\
&\qquad\qquad \leq 7370 - 2.02 \times 10^{-5} \cdot \text{Sal} - 146.3 \cdot \text{aTc}).
\end{aligned}
$$

Here, we include the contracts for each input condition and the transition between those conditions. A 3D visualization of the guarantees of all the top-level contracts is shown in Figure 4.2d. With the desired top-level system objectives written as contract specifications, we can propose component system designs and optimization over choice of components using the algebra on assume-guarantee contracts. Towards that end, we first build a library of parts of engineered biosensors using experimental data.

Modeling the specifications and constructing a library of sensors

First we represent a set of sensors using contracts. We build a library of sensor contracts using the experimental data for the sensors in the Marionette bacterial cell

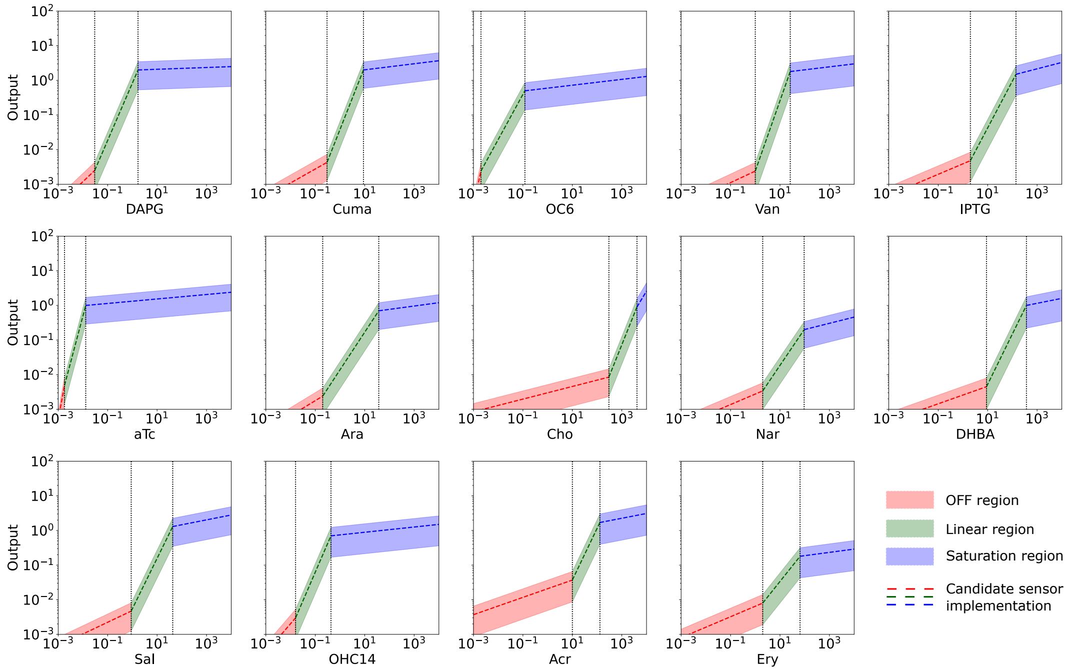


Figure 4.3: The assume-guarantee contracts for the Marionette sensor library. The three viewpoints for each sensor in the Marionette bacterial cell strain are shown in the shaded regions. The OFF region, described by C_s^{off}, is depicted in red. The linear region, described by C_s^{lin} is depicted in green. Finally, the saturation region, described by C_s^{sat}, is depicted in blue. The dashed lines show a possible candidate implementation for each sensor that satisfies the contracts. The first vertical dashed line in each plot shows the u_{start} value for each sensor and the second vertical dashed line shows u_{K}, the activation constant for the sensor.

strain [100]. Each sensor has three characteristic behaviors: 1) the off state, where the output stays close to zero (the non-zero expression in this state is termed as "leaky response"), 2) the linear rate of output, where the output responds linearly to the input (in log scale), and 3) the saturation state, where the output saturates to a maximum constant value. The three contracts for a sensor, s, with input u and output y are

$$C_s^{\text{off}} = (0.08u_{\text{start}} \leq u \leq u_{\text{start}}, m_1^{\text{off}}u + b_1^{\text{off}} \leq y \leq m_2^{\text{off}}u + b_2^{\text{off}}), \tag{4.2}$$

$$C_s^{\text{lin}} = (u_{\text{start}} \leq u \leq u_{\text{K}}, m_1^{\text{lin}}u + b_1^{\text{lin}} \leq y \leq m_2^{\text{lin}}u + b_2^{\text{lin}}), \tag{4.3}$$

$$C_s^{\text{sat}} = (u_{\text{K}} \leq u \leq 8u_{\text{K}}, m_1^{\text{sat}}u + b_1^{\text{sat}} \leq y \leq m_2^{\text{sat}}u + b_2^{\text{sat}}), \tag{4.4}$$

where m and b are the slopes and the intercepts of each affine relation respectively. The constant u_{start} is the trigger value at which sensor starts responding to the input, and u_{K} is the activation constant for the sensor. Each of these constants are computed using the Marionette sensor characterization experimental data for each sensor. Note that for the convenience of notation, we have removed the subscript s from each of the constants. We show the computation for one set of constants: $m_1^{\text{off}}, m_2^{\text{off}}$ and $b_1^{\text{off}}, b_2^{\text{off}}$. These are the slope and the intercept of the linear relations between the output y and the input u in the OFF region. We compute these as follows:

$$\begin{aligned} y_{\text{leak}}^1 &:= y_{\text{leak}} + \sigma y_{\text{leak}}, \quad & y_{\text{leak}}^2 &:= y_{\text{leak}} - \sigma y_{\text{leak}} \\ m_1^{\text{off}} &= \frac{0.1 y_{\text{leak}}^1 - y_{\text{leak}}^1}{0.08 u_{\text{start}} - u_{\text{start}}}, \quad & b_1^{\text{off}} &= y_{\text{leak}}^1 - m_1^{\text{off}} u_{\text{start}} \\ m_2^{\text{off}} &= \frac{0.1 y_{\text{leak}}^2 - y_{\text{leak}}^2}{0.08 u_{\text{start}} - u_{\text{start}}}, \quad & b_2^{\text{off}} &= y_2^{\text{leak}} - m_2^{\text{off}} u_{\text{start}} \end{aligned}$$

Here, y_{leak} is the experimental measurement of the leaky output for sensor s and σ is the standard deviation to allow for a range of behaviors in the implementation of the sensor contract. The minimum observable output value is arbitrarily chosen to be equal to $0.1 y_{\text{leak}}$ at the input value of $0.08 u_{\text{start}}$ and the maximum input value has been arbitrarily set at $8 u_{\text{K}}$.

The contracts for the three viewpoints for all of the 14 sensors are illustrated in Figure 4.3. The allowed range of behaviors are shaded and a candidate implementation of the sensors that would satisfy the contracts is also shown.

4.6 Contract quotient to find the specifications of missing parts

Suppose that we have chosen the two sensors (subsystems 1 and 2) from the library, and we are also given a desired top-level system contract C_{sys} that the system must meet. We use the quotient in Pacti to find the specification of the missing object: the dCas9 repression mechanism (subsystem 3). This design synthesis problem is described in Figure 4.4.

Using Sal to express the xRFP gRNA and aTc to express the dCas9, we have

$$\begin{aligned} C_{\text{Sal}}^0 =& (1.8 \leq \text{Sal} \leq 22.5, 0.0001808 + 4.709 \times 10^{-5} \cdot \text{Sal} \leq \text{xRFP} \wedge \\ & \text{xRFP} \leq 2.355 \times 10^{-5} + 0.0003.616 \cdot \text{Sal}), \\ C_{\text{Sal}}^1 =& (645 \leq \text{Sal} \leq 5160, \\ & 0.2502 + 7.656 \times 10^{-5} \text{Sal} \cdot \text{Sal} \leq \text{xRFP} \leq 1.921 + 0.0005879 \cdot \text{Sal}), \\ C_{\text{aTc}}^0 =& (0.000288 \leq \text{aTc} \leq 0.0036, 0.0001904 + 0.2826 \cdot \text{aTc} \leq \text{dCas9} \wedge \\ & \text{dCas9} \leq 2.261 \times 10^{-5} + 2.38 \cdot \text{aTc}), \\ C_{\text{aTc}}^1 =& (0.013 \leq \text{aTc} \leq 0.104, 0.1698 + 3.265 \cdot \text{aTc} \leq \text{dCas9} \leq 1.43 + 27.5 \cdot \text{aTc}), \end{aligned}$$

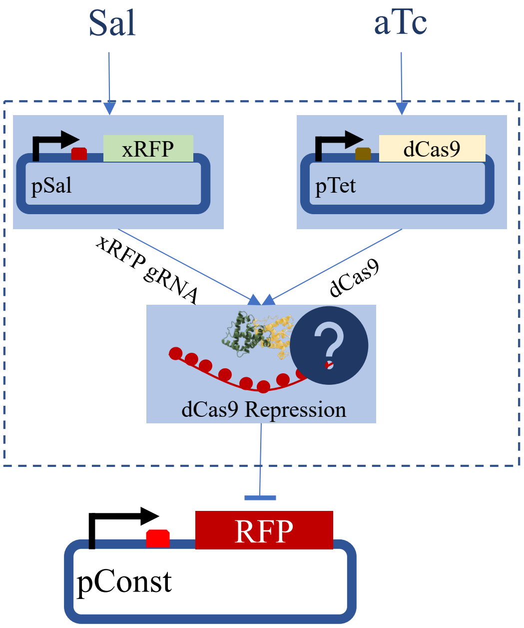


Figure 4.4: Finding the missing subsystem using quotient of contracts for synthesis of system designs. From the experimental implementation of the NAND logic with Sal-xRFP guide RNA and aTc-dCas9 together silencing constitutively expressed RFP, we find out the design specifications on the dCas9 repression mechanism.

here we have replaced the superscript “off” with “0” and “on” with “1” for brevity. For the top-level system, we use the C_{sys}^{0} and C_{sys}^{1} viewpoints defined in equation (4.1). Using the quotient, we obtain the inactive and the active viewpoints for the dCas9 repression subsystem, $C_{\text{dCas9}}^{\text{inactive}}$ and $C_{\text{dCas9}}^{\text{active}}$. The proposition that follows after the next Lemma guides the computation of the dCas9 repression contract, C_{dCas9}, as the quotient of the top-level contract from the composition of the sensor contracts in different viewpoints.

Lemma 3 (Preliminaries on Assume-Guarantee Contracts [153])**.** *We write the following relations between the binary operations on assume-guarantee contracts, C and C',*

$$C/C' = C \bullet (C')^{-1}, \tag{4.5}$$

where / is the quotient operator and $\bullet$ represents strong merging on contracts [153]. We also have

$$C \parallel C' = (C^{-1} \bullet C'^{-1})^{-1}. \tag{4.6}$$

where $\parallel$ represents contract composition. Finally, we note that conjunction and disjunction are related as

$$(C \wedge C')^{-1} = (C)^{-1} \vee (C')^{-1}. \tag{4.7}$$

Proposition 1. *The dCas9 repression mechanism specification is the conjunction of two quotient contracts that correspond to the dCas9 repression being active or inactive — (1) the quotient of the composition of the sensor contracts when they are "on" from the "on" top-level contract, and (2) the quotient of the composition of the sensor contracts when they are "off" from the "off" top-level contract. We have,*

$$\begin{aligned}
C_{dCas9} = &\left[C^0_{sys}/\left(C^0_{aTc} \parallel C^0_{Sal}\right) \wedge C^1_{sys}/\left(C^0_{aTc} \parallel C^0_{Sal}\right)\right] \\
&\vee \left[C^0_{sys}/\left(C^1_{aTc} \parallel C^0_{Sal}\right) \wedge C^1_{sys}/\left(C^1_{aTc} \parallel C^0_{Sal}\right)\right] \\
&\vee \left[C^0_{sys}/\left(C^0_{aTc} \parallel C^1_{Sal}\right) \wedge C^1_{sys}/\left(C^0_{aTc} \parallel C^1_{Sal}\right)\right] \\
&\vee \left[C^0_{sys}/\left(C^1_{aTc} \parallel C^1_{Sal}\right) \wedge C^1_{sys}/\left(C^1_{aTc} \parallel C^1_{Sal}\right)\right] \\
&\geq \left[C^0_{sys}/\left(C^0_{aTc} \parallel C^0_{Sal}\right)\right] \wedge \left[C^1_{sys}/\left(C^1_{aTc} \parallel C^1_{Sal}\right)\right] \\
&:= C^{inactive}_{dCas9} \wedge C^{active}_{dCas9}.
\end{aligned}$$

Proof. For the top-level system and the sensors, we have two contracts – the off and the on viewpoint, so we have

$$C_{\mathrm{s}} = C^0_{\mathrm{s}} \wedge C^1_{\mathrm{s}}, \tag{4.8}$$

where s is "sys" for the top-level system, "Sal" for the Sal sensor and "aTc" for the aTc sensor. For the dCas9 repression mechanism, we write,

$$\begin{aligned}
\bar{C}_{\mathrm{dCas9}} &= C_{\mathrm{sys}}/\left(C_{\mathrm{aTc}} \parallel C_{\mathrm{Sal}}\right), \\
&= C_{\mathrm{sys}}/\left((C^0_{\mathrm{aTc}} \wedge C^1_{\mathrm{aTc}}) \parallel (C^0_{\mathrm{Sal}} \wedge C^1_{\mathrm{Sal}})\right).
\end{aligned}$$

Using relation in equation (4.5), we write,

$$\bar{C}_{\mathrm{dCas9}} = C_{\mathrm{sys}} \bullet \left((C^0_{\mathrm{aTc}} \wedge C^1_{\mathrm{aTc}}) \parallel (C^0_{\mathrm{Sal}} \wedge C^1_{\mathrm{Sal}})\right)^{-1}.$$

Now, using equation (4.6), we can write

$$\begin{aligned}
\bar{C}_{\mathrm{dCas9}} &= C_{\mathrm{sys}} \bullet \left[(C^0_{\mathrm{aTc}} \wedge C^1_{\mathrm{aTc}})^{-1} \bullet (C^0_{\mathrm{Sal}} \wedge C^1_{\mathrm{Sal}})^{-1}\right]. \\
&= C_{\mathrm{sys}} \bullet \left[\left((C^0_{\mathrm{aTc}})^{-1} \vee (C^1_{\mathrm{aTc}})^{-1}\right) \bullet \left((C^0_{\mathrm{Sal}})^{-1} \vee (C^1_{\mathrm{Sal}})^{-1}\right)\right],
\end{aligned}$$

using the relation between conjunction and disjunction in equation (4.7). We distribute the merging operation $\bullet$ over $\vee$ to write,

$$\begin{aligned}
\bar{C}_{\text{dCas9}} &= C_{\text{sys}} \bullet \Big[\Big(\Big((C^0_{\text{aTc}})^{-1} \vee (C^1_{\text{aTc}})^{-1} \Big) \bullet (C^0_{\text{Sal}})^{-1} \Big) \\
&\qquad\qquad\qquad \vee \Big(\Big((C^0_{\text{aTc}})^{-1} \vee (C^1_{\text{aTc}})^{-1} \Big) \bullet (C^1_{\text{Sal}})^{-1} \Big) \Big], \\
&= C_{\text{sys}} \bullet \Big[\Big((C^0_{\text{aTc}})^{-1} \bullet (C^0_{\text{Sal}})^{-1} \Big) \vee \Big((C^1_{\text{aTc}})^{-1} \bullet (C^0_{\text{Sal}})^{-1} \Big) \\
&\quad \vee \Big((C^0_{\text{aTc}})^{-1} \bullet (C^1_{\text{Sal}})^{-1} \Big) \vee \Big((C^1_{\text{aTc}})^{-1} \bullet (C^1_{\text{Sal}})^{-1} \Big) \Big], \qquad (4.9) \\
&\geq C_{\text{sys}} \bullet \Big[\Big((C^0_{\text{aTc}})^{-1} \bullet (C^0_{\text{Sal}})^{-1} \Big) \vee \Big((C^1_{\text{aTc}})^{-1} \bullet (C^1_{\text{Sal}})^{-1} \Big) \Big].
\end{aligned}$$

In the last step, since quotient gives us the largest specification of the missing component that will meet the top-level specification, we can refine the quotient to remove the cross-terms. Converting the merge operations to composition and quotient using equations (4.5) and (4.6), we get

$$\begin{aligned}
\bar{C}_{\text{dCas9}} &\geq C_{\text{sys}} \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right)^{-1} \vee \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right)^{-1} \right] \\
&= C_{\text{sys}} \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right) \wedge \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right) \right]^{-1} \qquad (4.10) \\
&= C_{\text{sys}} / \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right) \wedge \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right) \right].
\end{aligned}$$

From equation (4.8), for the desired top-level specification, we have that

$$C_{\text{sys}} = C^0_{\text{sys}} \wedge C^1_{\text{sys}}.$$

So, we can further simplify the quotient,

$$\begin{aligned}
\bar{C}_{\text{dCas9}} &\geq \left(C^0_{\text{sys}} \wedge C^1_{\text{sys}} \right) / \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right) \wedge \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right) \right] \\
&= \left(C^0_{\text{sys}} \wedge C^1_{\text{sys}} \right) \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right) \wedge \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right) \right]^{-1} \\
&= \left(C^0_{\text{sys}} \wedge C^1_{\text{sys}} \right) \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right)^{-1} \vee \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right)^{-1} \right],
\end{aligned}$$

and since $\bullet$ distributes over $\wedge$, we have,

$$\begin{aligned}
\bar{C}_{\text{dCas9}} &\geq C^0_{\text{sys}} \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right)^{-1} \vee \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right)^{-1} \right] \\
&\qquad\qquad \wedge C^1_{\text{sys}} \bullet \left[\left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right)^{-1} \vee \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right)^{-1} \right] \\
&\geq C^0_{\text{sys}} \bullet \left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right)^{-1} \wedge C^1_{\text{sys}} \bullet \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right)^{-1} \qquad (4.11) \\
&= C^0_{\text{sys}} / \left(C^0_{\text{aTc}} \parallel C^0_{\text{Sal}} \right) \wedge C^1_{\text{sys}} / \left(C^1_{\text{aTc}} \parallel C^1_{\text{Sal}} \right) \\
&:= C^{\text{inactive}}_{\text{dCas9}} \wedge C^{\text{active}}_{\text{dCas9}},
\end{aligned}$$

where we define

$$C_{\text{dCas9}} := C_{\text{dCas9}}^{\text{inactive}} \wedge C_{\text{dCas9}}^{\text{active}}$$

with

$$C_{\text{dCas9}}^{\text{inactive}} = C_{\text{sys}}^{0} / \left(C_{\text{aTc}}^{0} \parallel C_{\text{Sal}}^{0} \right) \tag{4.12}$$

and

$$C_{\text{dCas9}}^{\text{active}} = C_{\text{sys}}^{1} / \left(C_{\text{aTc}}^{1} \parallel C_{\text{Sal}}^{1} \right). \tag{4.13}$$

To arrive at the full expression for $\bar{C}_{\text{dCas9}}$ as in the statement of the proposition, we can use the expression in equation (4.9) before refinement and expand similar to the steps in equation (4.11). □

Given the choice of sensors and the top-level contract, the contracts for the dCas9 mechanism are given by

$$\begin{aligned} C_{\text{dCas9}}^{\text{inactive}} = \big(&0.000306 \le \text{dCas9} \le 0.007452 \wedge 0.0002926 \le \text{xRFP} \le 0.007027\,, \\ &4212 - 5.619 \cdot \text{dCas9} - 7335 \cdot \text{xRFP} \le \text{RFP} \le 7325 - 33.82 \cdot \text{dCas9} \\ &-4.362 \times 10^{4} \cdot \text{xRFP}\big) \end{aligned} \tag{4.14}$$

$$\begin{aligned} C_{\text{dCas9}}^{\text{active}} = \big(&0.2964 \le \text{dCas9} \le 2.195 \wedge 0.3891 \le \text{xRFP} \le 2.769\,, \\ &3878 - 0.006968 \cdot \text{dCas9} - 1616 \cdot \text{xRFP} \le \text{RFP} \le 1250 - 0.2114 \cdot \text{xRFP} \\ &-4.613 \times 10^{-6} \cdot \text{dCas9}\big)\,. \end{aligned} \tag{4.15}$$

The contract for the dCas9 repression mechanism can be refined further by relaxing the assumptions and constraining the guarantees. The two sensor outputs feed the inputs of the dCas9 repression, hence, refining the dCas9 repression contract allows a wider range of sensor output measurements. We write a refined contract to allow more sensor choices from our constructed library:

$$\begin{aligned} C_{\text{dCas9}}^{\text{active}} = \big(&0.2964 \le \text{dCas9} \le 2.195 \wedge 0.26 \le \text{xRFP} \le 2.889\,, \\ &3770 - 0.006968 \cdot \text{dCas9} - 1616 \cdot \text{xRFP} \le \text{RFP} \le 1000 - 0.2114 \cdot \text{xRFP} \\ &-4.613 \times 10^{-6} \cdot \text{dCas9}\big)\,. \end{aligned} \tag{4.16}$$

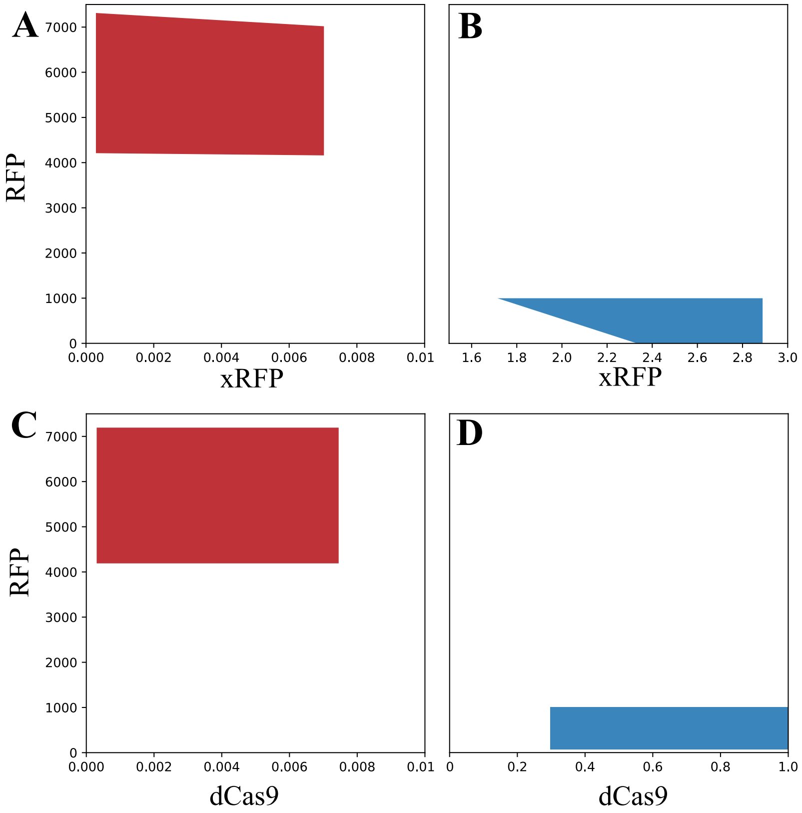


Figure 4.5: Contract guarantees of the dCas9 repression mechanism. (a) The dCas9 inactive contract, and (b) the dCas9 active contract guarantees for fixed dCas9 values and varying xRFP. (c) The dCas9 inactive contract, and (d) the dCas9 active contract guarantees for fixed xRFP values and varying dCas9.

This resulting contract for subsystem 3 guarantees that it represses the RFP level dependent on its inputs, xRFP and dCas9. We can provide this missing-component contract to an expert for independent implementation. The guarantees for this synthesized contract are visualized in Figure 4.5.

Contract composition and sensor selection

Now that we have the missing dCas9 repression mechanism specification, we can use it to optimize the choice of sensors by exploring the design space. We find the optimal choice of sensor to use from the library of 14 sensors that we constructed earlier. To optimize, we consider the commonly desired metric in synthetic biology — the fold-change of the output measurement.

We choose and fix the aTc sensor for our subsystem 2 (s_2 in Figure 4.6a). With C_{dCas9} available, we can go through the library of remaining 13 sensors to choose the sensing behavior for subsystem 1 (s_i in Figure 4.6a). Using a contract composition-based strategy, we can predict the fold-change of the RFP level for all choices of sensors. Finally, we predict the sensor that maximizes the top-level fold change. This methodology is illustrated in Figure 4.6.

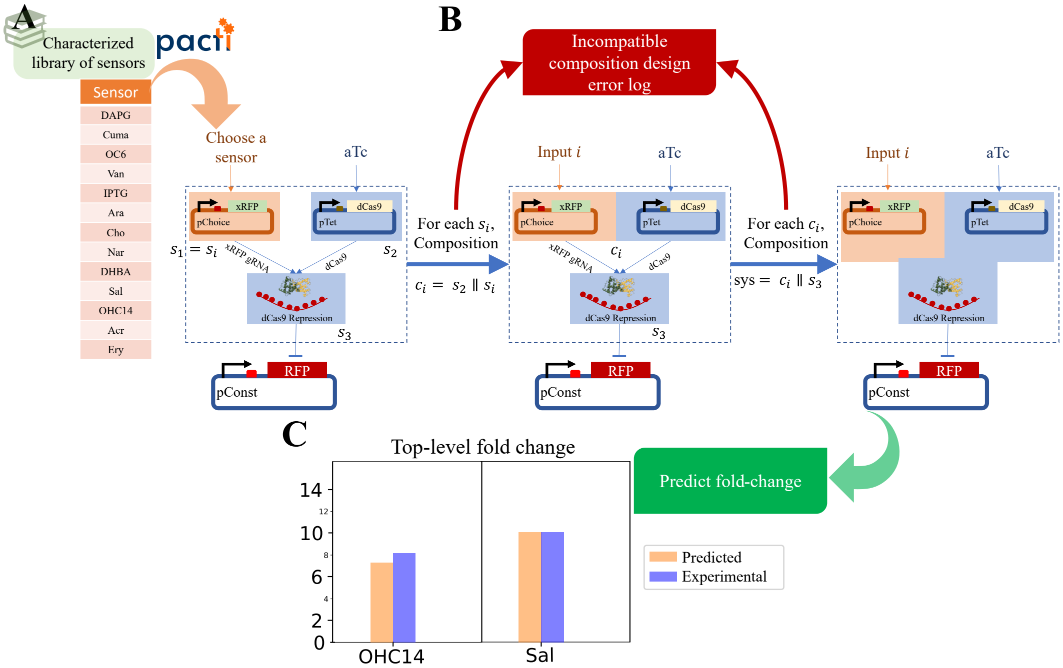


Figure 4.6: Prediction and validation of system fold-change by design-space exploration using a contract composition-based strategy. (a) We use the library of sensor contracts that we have developed in Pacti to choose the first subsystem (s_1) in the NAND gate design (marked in orange). We iterate over the library of sensors by choosing the sensor s_i to use as the first subsystem. (b) First, we compose s_i with s_2 (the fixed aTc sensor) to obtain c_i. Then, we compose the dCas9 repression mechanism contract (s_3) with c_i. Pacti captures any incompatibility related design errors in these composition computations. These design errors are stored (shown in red box). In this way, some sensor choices in the library are rejected. Finally, (c) we predict the fold-change of the system with different sensor choices and compare with the fold-change obtained experimentally.

Our prediction strategy corresponds to the experimental implementations of this biological circuit with the aTc as the fixed sensor that outputs dCas9. We implemented the NAND biological circuit as shown in Figure 4.2 and tested two different sensor choices for subsystem 1 – Sal and OHC14. As discussed above, we have used the data for the Sal sensor to compute the quotient for the missing specification of the dCas9 repression mechanism. The output measurements of RFP when OHC14 sensor was used instead of Sal has not been used. Hence, the OHC14 data acts as a good testing dataset for our prediction.

The contract for aTc, C_{aTc}, is given in three viewpoints as described above by using $K = 0.013, y_{\max} = 1, \text{start} = 0.0036$, and $y_{\text{leak}} = 4.9 \times 10^{-3}$. For subsystem 3, we use the dCas9 active and inactive contracts derived using the quotient in equations (4.12) and (4.16). For each sensor in the library with contract C_{s_i}, we use Pacti to compute the system-level contract by composing the chosen sensor contract, with the available subsystem contracts: $C_{\text{sys}} = C_{\text{aTc}} \parallel C_{s_i} \parallel C_{\text{dCas9}}$. When computing this composition for some of the sensors in the library, Pacti returns the error "unsatisfiable in the given context." This means that the guarantees of this sensor are insufficient to meet the assumptions of the component to which it drives outputs (the dCas9 repressor subsystem). Thus, Pacti allows us to identify potential design errors. Using the active dCas9 repression mechanism specification in equation (4.16), we get that the sensors DAPG, Cuma, Van, Ara, Nar, Acr, and Ery lead to a design error. Hence, these sensors choices are rejected. For example, for DAPG, we get the following output from Pacti,

```
The elimination of variables
[‘dCas9’, ‘xRFP’, ‘RFP’]
by refining terms
[
  dCas9 <= 2.195
  -dCas9 <= -0.2964
  xRFP <= 2.889
  -xRFP <= -0.26
]
in context
[
  -aTc <= -0.013
  aTc <= 0.104
  -DAPG <= -1.7
  DAPG <= 13.6
  -26.39 aTc + dCas9 <= 1.372
  4.384 aTc - dCas9 <= -0.228
  -0.07227 DAPG + xRFP <= 3.317
  0.01177 DAPG - xRFP <= -0.5401
] was not possible.
```

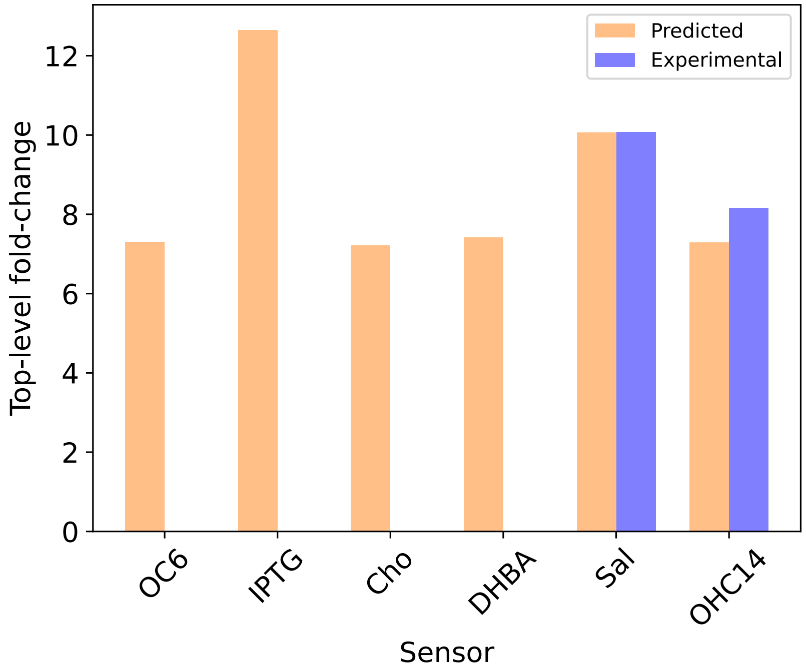


Figure 4.7: The prediction of fold-change of the NAND logic biological circuit. Each sensor in the library has a different response function that is captured in the assume-guarantee contracts. Some choices out of the 13 available sensors are rejected as they are incompatible with the desired system specifications. We show the fold-change computed from the experimental implementation (one replicate in an engineered Marionette strain of E. coli) of the circuit alongside the prediction using Pacti.

This suggests that the xRFP guide RNA levels produced by using the DAPG sensor were not sufficient to activate the dCas9 repression mechanism at the desired level. Hence, this sensor choice is rejected. Figure 4.6b visually describes this approach.

We predict the system behavior using the sensor choices that yield valid compositions. We use the fold change of the system $F = \mathrm{RFP}_{\mathrm{on}}/\mathrm{RFP}_{\mathrm{off}}$ as a performance criterion and predict the fold-change for each choice of sensor. We also predict the sensor choice that maximizes the fold-change. The predicted fold-change for all sensors is shown in Figure 4.7. From this results, we predict that the highest fold-change for the NAND gate can be achieved by using the “IPTG” sensor.

We validate our predictions using the experimentally observed fold-changes for the Sal and the OHC14 sensors. The comparison of the experimental and the predicted fold-change is shown in Figure 4.6c. We observe that for the Sal sensor, the predicted fold-change is close to the experimentally observed fold-change of RFP.

This is expected since we used the Sal sensor data to find the specification of the dCas9 repression mechanism. However, we did not use the data for OHC14 but even for this sensor choice, our predicted RFP fold-change is quite close to the experimental observation. It is important to note here that in this contract-based design approach, we did not model the explicit details of any of the sensors. Instead, we simply used the input-output characterization of each component to characterize the input-output response of the dCas9 repression mechanism. Using this synthesized specification, we predict the behavior of the system under various choices and the computational results are close to the experimental observations.

4.7 From specifications to implementations

In this section, we consider an example of a biological AND gate to demonstrate the use of contracts to develop a methodology that lets designers write specifications as assume-guarantee contracts and then systematically obtain implementations of the contracts. We also show how the contract-based design approach that we have discussed until now for static predictions can also be extended to model dynamical aspects of systems.

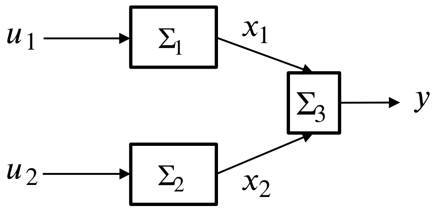


Figure 4.8: Composition of three subsystem models to achieve AND logic gate implementation in an engineered biological system.

First, we provide an overview of the system we are constructing, a biological AND gate. We introduce the contracts of the components comprising the system using two viewpoints: one for functionality and one for timing. Then we compose the contracts to obtain the specification of the entire system by using the properties of the components. We compose the models of each contract to verify that this composition meets the top-level specification we obtained. Then, starting from a top-level specification of the system and the specification of a partial implementation, we obtain the specification of a missing component that enables us to meet the top-level specification. We synthesize a model from this specification, and we show that the resulting composition of models implements the top-level specification.

Consider the design of an AND gate with inputs u_1 and u_2 and one output y. A biological circuit design to achieve the AND logic has been built and validated in [155]. This design is a composition of three subsystems that interact, as shown in Figure 4.8. In [155], the inputs u_1 and u_2 are chemical inducer signals (salicylate and arabinose, respectively), and the output y is the fluorescence of green fluorescent protein (GFP). Let the output of subsystem Σ_1 be x_1 that models the expression of an amber suppressor tRNA called supD in [155]. Similarly, the output x_2 of Σ_2 models the transcription of the mRNA that codes for the T7 RNA polymerase enzyme. However, the translation of the mRNA (x_2) is only possible when x_1 is also present. When both x_1 and x_2 are present, the translation of the T7 RNA polymerase can occur, which activates the T7 promoter in Σ_3 that codes for a fluorescent protein GFP, the output y of the system. The top-level specification for the AND gate is given in Table 4.1.

Specifications for timing constraints

We can write the specification for the subsystem Σ_1 by describing the assumptions and guarantees of the biological design. We assume that at time $t = \tau_{u_1}$ we have $u_1 \geq u_{1_{\min}}$, and u_1 stays over this threshold. The contract $C_1 = (a_1, g_1)$ for Σ_1 guarantees that $x_1 \geq x_{1_{\min}}$ at time $t \leq \tau_{u_1} + t_1$. Here u_1 is the salicylate inducer concentration, and x_1 is the expression level of the supD gene that is downstream of the pSal promoter.

We split our specification in two viewpoints. First, we have a functionality viewpoint that says that $x_1 \geq x_{1_{\min}}$ follows from $u_1 \geq u_{1_{\min}}$. The other is a timing viewpoint that says that the event τ_{x_1}, defined as the time when $x_1 \geq x_{1_{\min}}$, happens at most t_1

u_1	u_2	y
0	0	y_ϵ
$u_{1_{\min}}$	0	y_ϵ
0	$u_{2_{\min}}$	y_ϵ
$u_{1_{\min}}$	$u_{2_{\min}}$	y^*

For $u_1 \geq u_{1_{\min}}, u_2 \geq u_{2_{\min}}$:

$$y^* \geq F y_\epsilon, \qquad t \leq \tau_y$$

Table 4.1: The static AND gate specifications are such that when both inputs u_1 and u_2 are greater than their specified minimum values, we have $y^* \geq F y_\epsilon$, where $F > 1$ is the desired fold change in output compared to the leaky output y_ϵ. The dynamic specifications add that the output achieves the desired fold-change in time $t \leq \tau_y$.

time units after the event τ_{u_1}, defined as the time when $u_1 \geq u_{1_{\min}}$. That is, we have the following two contract viewpoints:

$$\begin{aligned} C_1^f &= (u_1 \geq u_{1_{\min}}, x_1 \geq x_{1_{\min}}) \text{ and} \\ C_1^t &= (1, \tau_{x_1} \leq \tau_{u_1} + t_1), \end{aligned}$$

where 1 in the assumptions of the last contract represents the top element of the constraint lattice, i.e., the Boolean value "true."

For Σ_2, we have the input u_2 (arabinose) that activates the pBAD promoter to express the T7Ptag gene downstream. For this subsystem, if we assume that at $t = \tau_{u_2}$, u_2 crosses the threshold $u_2 \geq u_{2_{\min}}$, then the subsystem specification guarantees that $x_2 \geq x_{2_{\min}}$ at time $t \leq \tau_{u_2} + t_2$. The functionality and timing contracts $C_2 = (a_2, g_2)$ for Σ_2 are

$$\begin{aligned} C_2^f &= (u_2 \geq u_{2_{\min}}, x_2 \geq x_{2_{\min}}) \text{ and} \\ C_2^t &= (1, \tau_{x_2} \leq \tau_{u_2} + t_2), \end{aligned}$$

where τ_{x_2} is, as before, the event when x_2 crosses the threshold $x_2 \geq x_{2_{\min}}$.

For Σ_3, we have inputs x_1, the tRNA supD, and x_2, the engineered T7 RNA polymerase transcript from Σ_1 and Σ_2, respectively. The translation of T7 RNA polymerase occurs only in presence of both x_1 and x_2 that then drives the production of the output, y, the green fluorescent protein. Under the assumptions that $x_1 \geq x_{1_{\min}}$ and $x_2 \geq x_{2_{\min}}$ starting at some $t = \max(\tau_{x_1}, \tau_{x_2})$, Σ_3 guarantees that the output y is at least $F > 1$ fold-change higher than the leaky expression output y_ϵ at time $\tau_y \leq t_3 + \max\{\tau_{x_1}, \tau_{x_2}\}$. Hence, the contracts for Σ_3 are

$$\begin{aligned} C_3^f &= (x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}}, y \geq F y_\epsilon) \text{ and} \\ C_3^t &= (1, \tau_y \leq \max\{\tau_{x_1}, \tau_{x_2}\} + t_3). \end{aligned}$$

For all of these contracts, we also write the contract C_i^{off} for the condition where the input $u_i \leq u_{i_{\min}}$ and similarly for the timing viewpoint. These contracts can be constructed similar to the approach discussed in the previous section for the NAND gate.

Generating specifications of the system

Now that we have the specifications for the three elements of the system, we seek the specification of the entire system. First, we use the operation of composition to obtain the specification of the subsystem consisting of components Σ_1 and Σ_2. So, using Pacti, we can compute $C^f_{123} = C^f_1 \parallel C^f_2 \parallel C^f_3$ and $C^t_{123} = C^t_1 \parallel C^t_2 \parallel C^t_3$. We get,

$$\begin{aligned} C^f_{123} &= \left((u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}}), y \geq F y_\epsilon\right), \\ C^t_{123} &= (1, \tau_y \leq \max\{\tau_{u_1} + t_1, \tau_{u_2} + t_2\} + t_3). \end{aligned}$$

These contracts give us a specification for the entire system. They only refer to variables that lie at the interface between the system and its environment, namely u_1, u_2, and y; there is no mention of x_1 and x_2. This allows us to "black-box" the system so that it can be used as a component of a more involved system.

From specifications to dynamical models

We explore the link between the contract specifications and the component models as differential equations. As discussed above, the formal specifications describe the desired functional and timing objectives for the system. By developing dynamical models with biological details from these specifications, we can map the desired objectives to system implementations. We first develop first-order dynamical models by employing standard functions to convert time delays and function gain to dynamical equations. Then we add more detail to these models. A flow chart summarizing this process is shown in Figure 4.9.

For Σ_1, we can write the following equation:

$$x_1(t) = x_1(0) + k_1 \left(1 - e^{-\frac{t-\tau_{u_1}}{t_1}}\right) \cdot s(t - \tau_{u_1}),$$

where $s(\cdot)$ is the step function such that for $t < \tau_{u_1}$, $x_1(t) = x_1(0)$ and for $t \geq \tau_{u_1}$, $x_1(t) = x_1(0) + k_1(1 - e^{-(t-\tau_{u_1})/t_1})$. The parameter k_1 is given according to the specification in the contract as

$$k_1 = \frac{e(x_{1_{\min}} - x_1(0))}{e - 1},$$

so that at $t = \tau_{x_1}$, we have $x(t) = x_{1_{\min}}$. Note that this is an implementation of the contract C_1 since it guarantees that $x_1(t) \geq x_{1_{\min}}$ when the assumption $u_1 \geq u_{1_{\min}}$ is satisfied. Further, the timing contract is also satisfied as the implementation guarantees that $x_1 \geq x_{1_{\min}}$ at time $t \geq \tau_{x_1}$. The input is modeled as an ideal activator function by using the step function. Following a similar approach for Σ_2, we have

$$x_2(t) = x_2(0) + k_2 \left(1 - e^{-\frac{t-\tau_{u_2}}{t_2}}\right) \cdot s(t - \tau_{u_2}),$$

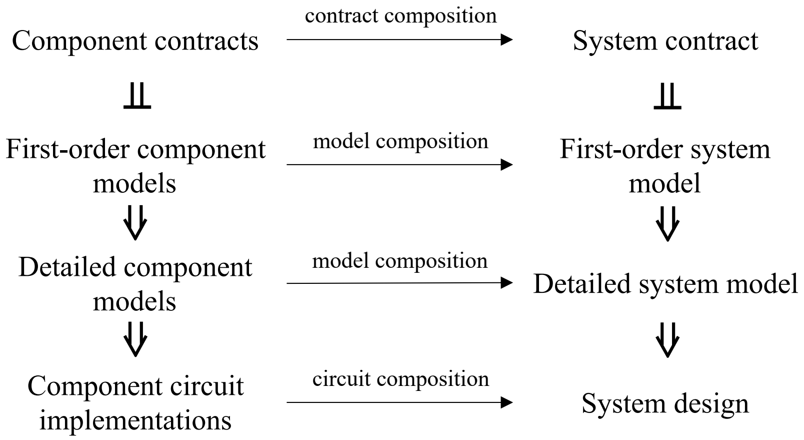


Figure 4.9: Contract-based design allows us to compute a system specification from the specifications of the system's components. The system-level contract has the property that its implementations are the compositions of the implementations of the component contracts. This property holds even if we keep adding detail to our models.

where

$$k_2 = \frac{e(x_{2_{\min}} - x_2(0))}{e-1}.$$

Finally, for Σ_3, define $\tau_x := \max(\tau_{x_1}, \tau_{x_2})$. We have

$$y = y_\epsilon + k_3\left(1 - e^{-\frac{t-\tau_x}{t_3}}\right) \cdot s(t - \tau_x),$$

where

$$k_3 = \frac{e(F-1)y_\epsilon}{e-1}.$$

We now derive differential equations that will enable us to make a clearer connection between the first-order models and models that are more descriptive of biological implementations. We differentiate the equations for x_1, x_2, and y with respect to time to obtain

$$\begin{aligned} \dot{x}_1 &= \left(k_{x_1} - d_{x_1}x_1\right) \cdot s(t - \tau_{u_1}), \\ \dot{x}_2 &= \left(k_{x_2} - d_{x_2}x_2\right) \cdot s(t - \tau_{u_2}), \\ \dot{y} &= \left(k_y - d_y y\right) \cdot s(t - \tau_x), \end{aligned} \tag{4.17}$$

where

$$\begin{aligned} k_{x_1} &= \frac{k_1 + x_1(0)}{t_1} = \frac{ex_{1_{\min}} - x_1(0)}{(e-1)t_1}, \quad d_{x_1} = \frac{1}{t_1}, \\ k_{x_2} &= \frac{k_2 + x_2(0)}{t_2} = \frac{ex_{2_{\min}} - x_1(0)}{(e-1)t_2}, \quad d_{x_2} = \frac{1}{t_2}, \\ k_y &= \frac{k_3 + y_\epsilon}{t_3} = \frac{(eF-1)y_\epsilon}{(e-1)t_3}, \quad d_y = \frac{1}{t_3}. \end{aligned}$$

In the ODE model, for $t < \tau_{u_1}$ we have $\dot{x}_1 = 0$, so we set the boundary condition as $x_1(t) = x_1(0)$ for all $t < \tau_{u_1}$. Similarly, we write the initial conditions as boundary values for $x_2(t)$ and $y(t)$.

Remark: The dynamical model shown above follows from the component specifications written as contracts. Hence, the parameters in the models are functions of the component specifications. Numerical simulations of the dynamical models are shown in Figure 4.10. The simulations show that the composition of all system models is a valid implementation of the system-level contract specification we computed.

Observe that in the first expression of equation (4.17), for all $t \in [0, \tau_{u_1})$, we have $\dot{x}_1 = 0$ and for all $t \in [\tau_{u_1}, \infty)$, $\dot{x}_1 = k_{x_1} - d_{x_1}x_1$. If the initial condition $x_1(0) = 0$, then we can simplify the differential equation to $\dot{x}_1 = k_{x_1}s(t - \tau_{u_1}) - d_{x_1}x_1$ since the step function multiplied with the linear degradation term is redundant. We can follow a similar procedure for all equations in (4.17) to write ODE models such that the inputs to each subsystem activates its response through the production term. Under the assumption that the initial conditions are $x_1(0) = x_2(0) = y(0) = 0$, we have

$$\begin{aligned}
\dot{x}_1 &= k_{x_1} \cdot s(t - \tau_{u_1}) - d_{x_1}x_1 \\
\dot{x}_2 &= k_{x_2} \cdot s(t - \tau_{u_2}) - d_{x_2}x_2 \\
\dot{y} &= k_y \cdot s(t - \tau_x) - d_y y.
\end{aligned} \tag{4.18}$$

The composition of subsystems gives us an implementation of the top-level system contract as

$$\dot{y} = k_y \cdot s(\max(\tau_{u_1} + t_1, \tau_{u_2} + t_2)) - d_y y$$

by using the contract definitions.

We have shown the synthesis of dynamical models for the subsystems and their composition from the specification contracts. However, these first-order models are not descriptive of biological implementations. As a result, it is not possible to map the design objectives to the controllable parameters in the implementation. In order to expand the model to include details of biological mechanisms, we observe that the first-order models use an ideal activation function — the step function. A common activation function used in the mathematical modeling of biological systems is a Hill function. An activation Hill function (rate = $u^n/(u^n + K^n)$) models activation as slow growth (rate $\ll 1$) until a threshold (K) is reached, then, an ultra-sensitive response to asymptotically reach the maximum value (rate = 1). From our development of the

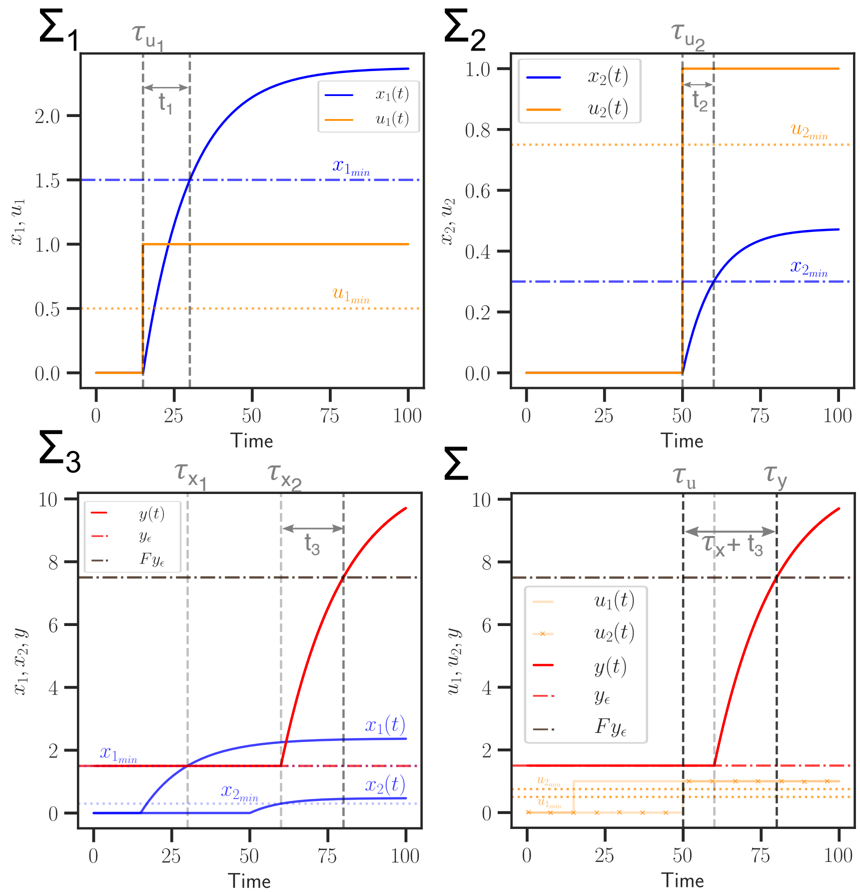


Figure 4.10: Simulations for each subsystem, Σ_i, $i = \{1, 2, 3\}$ and the composed system Σ are shown in the figure. Dotted and dashed lines represent the assumptions and guarantees, respectively, while the solid lines show an implementation of the contract.

mathematical models shown above, we can use a Hill function in place of the step function to model the non-ideal activation characteristics of an activatable promoter in the genetic circuit. An expanded model can be written as

$$\begin{aligned}
\dot{x}_1 &= k_{x_1} \frac{u_1^{n_1}}{u_1^{n_1} + u_{1_{\min}}^{n_1}} - d_{x_1} x_1 \\
\dot{x}_2 &= k_{x_2} \frac{u_2^{n_2}}{u_2^{n_2} + u_{2_{\min}}^{n_2}} - d_{x_2} x_2 \\
\dot{y} &= k_y \frac{x_1^{n_{x_1}}}{x_1^{n_{x_1}} + x_{1_{\min}}^{n_{x_1}}} \frac{x_2^{n_{x_2}}}{x_2^{n_{x_2}} + x_{2_{\min}}^{n_{x_1}}} - d_y y.
\end{aligned} \tag{4.19}$$

Note that in the limit of $n_1 \to \infty$, $n_2 \to \infty$, and $n_{x_1}, n_{x_2} \to \infty$, the respective Hill functions converge to a step function with the threshold τ variables defined according to the activation constants specified in the design: $u_{1_{\min}}, u_{2_{\min}}, x_{1_{\min}}$, and

$x_{2_{\min}}$. Hence, the model in equation (4.19) converges to the dynamical model in equation (4.18).

Remark: For a biological implementation, Hill function coefficients are usually constrained to $n_i \leq 4$, where i denotes the subscripts for n above. With this additional constraint, the Hill function model may not meet the guarantees even when the assumptions are satisfied. We can offset this by letting the first-order model satisfy the guarantees such that $y > Fy_\epsilon$ at $t \ll \tau_x + t_3$ so that even with a limited Hill coefficient, the detailed model can satisfy the stated guarantees in the contract.

Finally, we can expand the model in equation (4.19) to a chemical reaction network (CRN) model of the circuit implementation by using model reduction techniques such as conservation laws, state transformations, and time-scale separation, as discussed in [26], [34], [40], [112] and related papers. The expansion to CRN models is out of scope for this thesis and may be addressed in future research.

Design synthesis of missing subsystem

Similar to the NAND gate example, we can use the quotient operation on contracts to find the specification of a missing component in the system. Then, using the method above to synthesize ODE models from contracts, we can propose dynamical implementation model for a component that is missing in the system. Suppose that we are given a specification for the entire system:

$$\begin{aligned} C_s^f &= \left((u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}}), y \geq y_{\min}\right), \\ C_s^t &= \left(1, \tau_y \leq \max\{\tau_{u_1}, \tau_{u_2}\} + t_3\right). \end{aligned}$$

Suppose we also have available the specification of a subsystem, say the composition of Σ_1 and Σ_2 — C_{12}^f and C_{12}^t as computed above. The question is, what is the specification of an element that we have to add to C_{12}^f and to C_{12}^t so that the resulting implementation meets the system-level specifications, C_s^f and C_s^t? The largest specification with this property is given by the contract quotient. Using Pacti, we compute the quotient $\bar{C}_q^f = C_s^f / C_{12}^f$ and $\bar{C}_q^t = C_s^t / C_{12}^t$:

$$\bar{C}_q^f = \begin{pmatrix} u_1 \geq u_{1_{\min}} \wedge u_2 \geq u_{2_{\min}} \wedge \\ x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}}, \\ y \geq y_{\min} \end{pmatrix},$$

$$\bar{C}_q^t = \begin{pmatrix} \tau_{x_1} \leq \tau_{u_1} + t_1 \wedge \tau_{x_2} \leq \tau_{u_2} + t_2, \\ (\tau_y \leq \max\{\tau_{u_1}, \tau_{u_2}\} + t_3) \vee \\ \neg(\tau_{x_1} \leq \tau_{u_1} + t_1 \wedge \tau_{x_2} \leq \tau_{u_2} + t_2) \end{pmatrix}.$$

We can refine these two contracts by removing references to the inputs u_1 and u_2:

$$C_q^f = \begin{pmatrix} x_1 \geq x_{1_{\min}} \wedge x_2 \geq x_{2_{\min}}, \\ y \geq y_{\min} \end{pmatrix},$$

$$C_q^t = \begin{pmatrix} 1, \tau_y \leq \max\{\tau_{x_1} - t_1, \tau_{x_2} - t_2\} + t_3 \end{pmatrix}.$$

Observe that any implementation of this contract is guaranteed to satisfy the system-level specification when it operates in conjunction with an implementation of C_{12}. We now look for an implementation of C_q. We can propose a first-order model using the following expression:

$$y(t) = k_3 \left(1 - e^{-\frac{t-\tau_x}{t_3}}\right) \cdot s(t - \tau_x), \tag{4.20}$$

where we define $\tau_x := \max(\tau_{x_1} - t_1, \tau_{x_2} - t_2)$ and

$$k_3 = k_o \frac{e(y_{\min} - y(0))}{e - 1}.$$

The dynamical model is given by

$$\dot{y} = \left(k_y - d_y y\right) \cdot s(t - \tau_x),$$

where

$$k_y = \frac{k_3}{t_3} \quad d_y = \frac{1}{t_3}.$$

With $k_o = 1$, this model satisfies the guarantees such that $y = y_{\min}$ at the required timing guarantee $t = \tau_x + t_3$. This synthesized dynamical implementation model can be expanded further to include the modeling details specific to a synthetic biology implementation by using a Hill activator function (similar to model in (4.19)) instead of the step function. As discussed in the previous subsection, the biological implementation with a Hill function has an additional constraint on the Hill coefficient. To offset this, we set $k_o > 1$ so that the model in (4.20) satisfies $y > y_{\min}$ at a time $t < \tau_x + t_3$. In this way, the biological implementation with the Hill functions also satisfies the guarantees as shown in Figure 4.11.

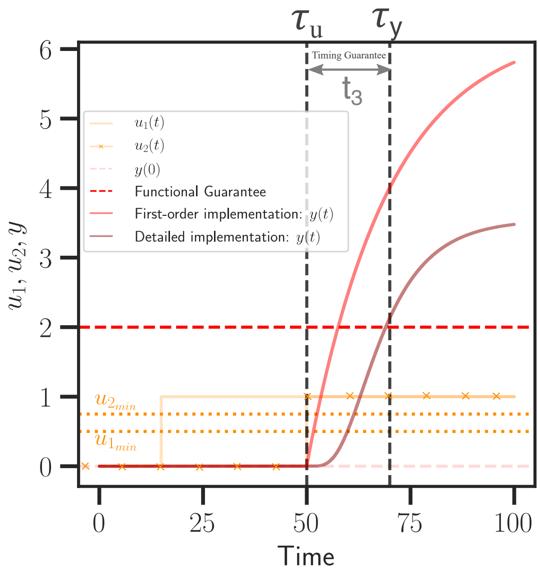


Figure 4.11: Synthesis of missing subsystem using the quotient operation of contracts. The dotted and dashed lines show the contract assumptions and guarantees, respectively, while solid lines show implementations. Here τ_u is defined as $\tau_u := \max\{\tau_{u_1}, \tau_{u_2}\}$.

4.8 Summary

We presented a contract-based design framework for synthetic biology. Current modeling practices in synthetic biology are limited to system analysis and inverse problems for system identification. Our results are a step towards a design framework that reasons about system properties using contracts and is capable of correlating implementations with specifications. We demonstrated a predictive design framework where we used contracts to predict the specifications of an internal biological mechanism. Using this specification, we predicted the performance of various design choices in a system design. We also showed how we could reason about system-level properties by composing the contracts of the components; then we composed the dynamical models for each component and verified that their composition satisfied the system-level contract. From the dynamical implementations, we derived detailed models that are closer to a biological implementation. With this approach, we can speed up the experimental library screening steps in circuit design and use subsystem specifications to predict compositional system behavior. The assume-guarantee contract-based design approach is fully general, so we expect to extend the results in this chapter to include context effects and assist parameter screening in biological circuit designs.

Chapter 5

FUTURE RESEARCH: SCALING MODEL-GUIDED DESIGN

5.1 The interplay between implementation and phenomenological models

An interesting future direction would be to extend the model reduction robustness results for the expression of multiple genes together to explore retroactivity [156] and its effects on various phenomenological models used for the design of such systems. This study would be a step in building towards a modular design framework [157] that considers the design of multiple modules together with their context-dependence. The contract-based design approach in Chapter 4 is one such modular and scalable design framework.

Representing large system models in phenomenological terms is a challenging task which would require going beyond QSSA based model reduction presented in Chapter 2 of this thesis. Hence, derivation of reduced models by introducing defined coordinate transformations for states and parameters might be worth investigating as well. Heuristic guidance is another approach that may be used for model reduction of large system models. The greedy algorithm in [66] is a possible direction to apply our results to large biological network models. More research on similar lines could improve the scalability of the tools discussed in Chapter 2. Finally, we would also like to note that to compute the robustness metric, new theoretical results on sensitivity of the model reduction error with respect to simultaneous multiple parameter co-variations could give new insights. Using sensitivities for robustness analysis is not as widely explored for nonlinear system analysis and hence building on the results in this thesis could lead to alternative robustness analysis methods.

On the other hand, from the design point of view, it is important to develop novel nonlinear model transformation algorithms that can transform phenomenological models to larger nonlinear models of a prescribed mathematical structure. Towards that end, it would be interesting to develop a computational method based on nonlinear ODE modeling and chemical reaction network theory that can transform a given nonlinear dynamical model into a desired coordinate structure while preserving specified properties.

On the lines of the preliminary results in Chapter 4, we could comprehensively develop "approximately equivalent" classes of nonlinear functions that are relevant to biological system design. Tunable granularity of these models under structural constraints provided by the contracts may provide a way towards platform-based design [149] for synthetic biology. Here, the assume-guarantee contracts would describe the design specifications such as the dynamic range, robustness to disturbance, time constants, or period of oscillations whereas the most detailed model serves as a circuit implementation guide for the biologist designing the circuits. Finally, integrating the results in Chapter 2 of this thesis would allow one to quantify the mapping between models of different granularity.

5.2 Modeling and simulation frameworks

There are two noteworthy limitations of the modeling framework presented in Chapter 3 of this thesis, both of which are also open research problems in systems and synthetic biology:

1. Scalable biological circuit design: Although the computational pipeline presented Chapter 3 applies to larger biological circuits as well, it is only demonstrated for circuits with 3-4 components. Various challenges limit our ability to quantify and model larger circuits. From the biological standpoint, it is still unclear how the context changes and affects the performance when more components are added to a system. On the other hand, parameter identification for higher dimensional problems is computationally inefficient, which presents another bottleneck in validating larger circuit models. Towards this end, integrating the results in Chapter 4 with the software pipeline presented in Chapter 3 might lead to fruitful new design tools for synthetic biology.

2. In this thesis, we have shown that reduced models can be systematically obtained from detailed models for computationally feasible parameter inference. But, in some cases, we observe that the reduced models are unable to fit the experimental data (for example, the plateauing of fluorescence as the cell-free extract runs out of resources). This requires further development of characterized library of parts that are utilized in synthetic biology. A well-characterized library of sensors was used in Chapter 4 to reliably predict system performance. Hence, future experimental and computational efforts on carefully characterizing biological circuit parts in varying contexts is crucial.

5.3 Scaling up biological circuit designs

The utility of our design methodology in Chapter 4 is apparent when scaling the analysis presented until this point to larger circuit designs.

Fold-change prediction for a larger circuit

In Chapter 4, we considered the design of a NAND logic biological circuit that was built out of two sensors from a library of 14 sensors and a dCas9-mediated repression subsystem for which we synthesized the specification. Using the synthesized specification of the dCas9-mediated repression, we predicted the fold-change of the system under different sensor choices. This system had a total of four components — the sensor expressing dCas9, the sensor expressing the guide RNA [131], [158], the dCas9-mediated repression mechanism, and the constitutive RFP expression. To demonstrate the scalable nature of our approach, we consider an expanded circuit design.

In the expanded circuit design, the guide RNA is expressed in two parts that bind together to create the functional guide RNA. The proposed circuit diagram is shown in Figure 5.1A. In a candidate implementation of this circuit, we have the IPTG sensor from the Marionette library that expresses one half of the xLuxR guide RNA and the OHC14 sensor expresses the other half. These two combine together to form the functional xLuxR guide RNA. As before, aTc is used to express dCas9, which then binds with the guide RNA to repress the LuxR regulator. The LuxR regulator in turn represses the GFP signal. So, when all three inputs (IPTG, OHC14, and aTc) are “on”, the GFP repression turns off, leading to high GFP expression. In this way, this circuit has an AND logic (data shown in Figure 5.1B). For this circuit, in total, we have 7 components.

The contract-based design approach discussed in Chapter 4 can be easily extended for this circuit design. We can construct the contracts for all sensors using the library of assume-guarantee contracts for the Marionette library. We also have the specification for the dCas9-mediated repression mechanism from the NAND gate circuit design. So, other than the gRNA binding mechanism (see Figure 5.1A), we have contracts for all other subsystems. Using quotient, we find a specification for this missing part. The top-level contract is constructed using the experimental implementation of the candidate circuit design. We construct the top-level contracts $C_{\text{sys}}^{\text{on}}$, $C_{\text{sys}}^{\text{off}}$ and a library of regulator contracts shown in Figure 5.1C. We have C_{dCas9}

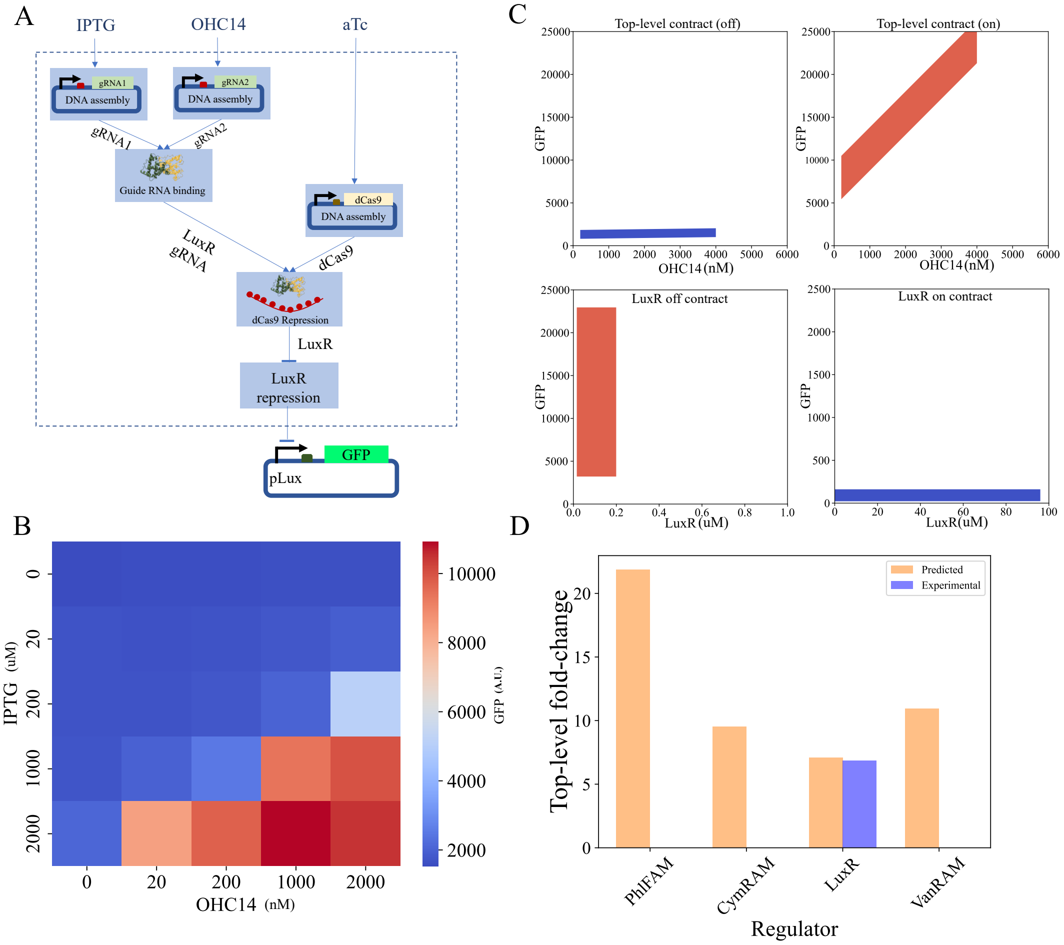


Figure 5.1: Application of contract-based design to a larger circuit design. (A) The circuit diagram for the dCas9-repression mechanism based three-input AND logic gate. A candidate implementation of this circuit is shown with IPTG, OHC14, aTc to repress LuxR, which represses GFP. (B) The experimental data for the circuit that is used to construct the top-level contracts for the system as IPTG and OHC14 is titrated at different concentrations (data courtesy of Shuwen (Eric) Lei, Caltech). (C) The top-level system contracts constructed using the experimental data and the contracts for the LuxR regulator constructeed using the data from the Marionette library. (D) The predicted fold-change with different choices of regulators. The LuxR prediction is shown alongside the experimental fold-change.

from the contract-based design of the dCas9-based NAND gate. Then, we compute the quotient for $C_{\text{gRNA-bind}}$ as

$$
\begin{aligned}
C^{\text{on}}_{\text{sensors}} &= C^{\text{on}}_{\text{IPTG}} \parallel C^{\text{on}}_{\text{OHC14}} \parallel C^{\text{on}}_{\text{aTc}}, \\
C^{\text{on}}_{\text{all}} &= C^{\text{on}}_{\text{sensors}} \parallel C^{\text{active}}_{\text{dCas9}} \parallel C^{\text{off}}_{\text{LuxR}}, \\
C_{\text{gRNA-bind}} &= C^{\text{on}}_{\text{sys}} / C^{\text{on}}_{\text{all}},
\end{aligned}
$$

where $C_{\text{dCas9}}^{\text{active}}$ is the active dCas9 repression mechanism contract and we use the $C_{\text{LuxR}}^{\text{off}}$ viewpoint for the LuxR regulator because the LuxR is repressed when the dCas9 is active. In turn, the output GFP signal is turned on when LuxR is off. The quotient gives the contract for the binding of gRNA 1 and gRNA 2 to form the functional xLuxR guide RNA. Following a similar process, we compute the contract $C_{\text{gRNA-unbind}}$ using the top-level contract when GFP is off, corresponding to the sensors (IPTG and OHC14) being off. We justified the use of this methodology with contract viewpoints earlier in Chapter 4.

Using this specification from quotient, we explore the choice of the regulator to use instead of LuxR to repress GFP. Similar to the NAND logic design, we compute the GFP fold-change by computing the composition of all subsystems to find the top-level contract in the on and the off conditions.

We explore the four regulator choices from the Marionette library (PhlFAM, CymRAM, LuxR, VanRAM) to avoid crosstalk with the IPTG, OHC14, and aTc sensors which are fixed choices. The fold-change prediction for the LuxR regulator is compared with the experimental data from the candidate circuit implementation (see Figure 5.1C. We observe that our fold-change prediction is close to the fold-change observed in the one replicate of the GFP fluorescence measurement in a Marionette strain of E. coli. If instead of LuxR, a different regulator is used to repress GFP, then we follow a similar process as above to compute the system fold-change. The prediction with different regulators is shown in Figure 5.1D.

A limitation of this approach is that the *in vivo* data is usually not calibrated to concentration units and reported in arbitrary units. With arbitrary units, the quantification of the input-output behavior in the assume-guarantee contracts needs to be re-calibrated for each circuit iteration. For example, we modified the contract for the dCas9-mediated repression that was synthesized in the NAND gate design so that it was compatible with the AND gate design. This is due to the change in the output fluorescence levels that were measured in the experiments. Similarly, we modified the assumptions of the regulators to predict the top-level fold-change for all four regulators. With the default values for the regulator, composition would not be compatible, so we focused on using the guarantees to compute the output fold-change. Standardized measurements and normalization in the contract-based design approach may address these issues. The specification abstraction in assume-guarantee contracts is a general framework that can be used to write contracts such that desired properties of the subsystems are the variables in the contracts instead of the measured signals. The experimental measurements may be used to compute those properties and the contract-based design framework can be applied on the analysis of the desired properties.

Large scale design-space exploration with Pacti

Finally, we discuss the scaling of the contract-based design approach from exploring only a few possibilities to a large number of design choices. We consider a hypothetical cascaded circuit schematic shown in Figure 5.2 that consists of a total of 7 subsystems. Four input sensors s_i may be chosen from the Marionette sensor library that we constructed in Chapter 4. The outputs from these four sensors feed two processor subsystems which then in turn feed another processor that produces the final output y. With four sensors to choose from a total of 14, we have a total of 1001 (14 choose 4) design possibilities for this system. With Pacti, we can quickly explore this design space by composing the system. We compute

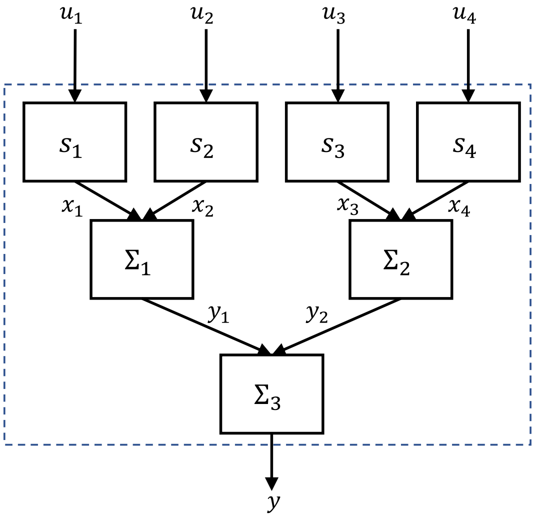


Figure 5.2: A hypothetical circuit design to demonstrate the scalability of our design methodology based on Pacti. We have four sensor choices $s_i, i \in [1, 2, 3, 4]$ from the library of 14 sensors which feed three subsequent processors, $\Sigma_i, i \in [1, 2, 3]$. The final system output is y.

$$C_1 = \left((C_{s_1} \parallel C_{s_2}) \parallel C_{\Sigma_1}\right)$$

$$C_2 = \left((C_{s_3} \parallel C_{s_4}) \parallel C_{\Sigma_2}\right)$$

$$C_{\text{sys}} = C_1 \parallel C_2 \parallel \Sigma_3.$$

On an AMD Ryzen Threadripper PRO 3955WX 16-Cores @ 3.8927 GHz and up to 32 threads, this space exploration ran in 4.4 seconds. A total of 82,428 composition operations were invoked. The smallest contract had 2 constraints and 2 variables, and the largest had 6 constraints and 6 variables. Out of all 1001 design possibilites hundreds of sensor choices are rejected as they are incompatible with the contracts for any of the Σ_i's. The result of this exploration is a list of successful design choices and a prediction of how each would perform. Given the choice of contracts for this *in silico* exploration, we obtained a total of 547 successful designs and all others were rejected.

We can also use a quotient-then-compose design strategy as before with this system. Suppose that we had experimental data for a fixed choice of $s_i, i \in [1, 2, 3, 4]$ and corresponding measurements of y. Also, assume that we have the specification for the composition of Σ_1 and Σ_2. This specification may be possible by using results from a previous partial experiment. Once we have these contracts, we can use quotient to find out the specification for Σ_3. Then, we can fix Σ_3, and explore optimal sensor choices or explore different specifications for Σ_1 or Σ_2. In this way, we can make reliable predictions for an experimental implementation of the system by learning from the experimental data for a partial implementation.

Speeding up the experimental design process

Usually, biological circuit design entails multiple experimental iterations to nail down controllable parameters such as promoter strengths, input levels, ribosome binding strengths, and physical conditions like pH and temperature. In our approach, we can explore the design possibilities using the input-output characteristics of each component to study the context effects and iteratively study larger system designs. Further, since the parameters of the implementation models are mapped to the system objectives, these controllable aspects in the experimental design can be manipulated accordingly. Hence, a formal design framework serves to minimize the experimental trial-and-error steps required to successfully design a biological circuit. Moreover, we can analyze how uncontrollable parameters corresponding to the internal biological mechanisms in the implementation relate to the system objectives.

Resource loading effects on system design

Engineered circuits with gene expression are dependent on cellular resources such as RNA polymerase, ribosome, ATP, and nucleic acids. For each subsystem to function as desired, a minimum level of these resources is required. At the same time, due to the resources being used by a subsystem, loading or retroactivity effects are commonly observed experimentally. Various system analysis approaches in the literature have analyzed retroactivity using mathematical models [156], [159], [160]. The contract-based design framework that we proposed in Chapter 4 can be easily scaled to describe such environmental assumptions as well. For example, consider RNA polymerase, ribosome, and ATP as three resources required for gene expression. We can add a resource viewpoint, C^r, to the contracts for the subsystems in addition to the timing, C^t, and the functionality, C^f, viewpoints. The resource viewpoint contract for subsystem Σ_1 can be written as

$$C_1^r = \begin{pmatrix} r_{\text{in}}^{\text{P}} > r_{1_{\text{min}}}^{\text{P}} \wedge r_{\text{in}}^{\text{R}} > r_{1_{\text{min}}}^{\text{R}} \wedge r_{\text{in}}^{\text{A}} > r_{1_{\text{min}}}^{\text{A}}, \\ (r_1^{\text{P}} = r_{1_L}^{\text{P}}) \wedge (r_1^{\text{R}} = r_{1_L}^{\text{R}}) \wedge (r_1^{\text{A}} = r_{1_L}^{\text{A}}) \end{pmatrix}.$$

This contract states that if the input resources for polymerase (r_{in}^{P}), ribosome (r_{in}^{R}), and ATP (r_{in}^{A}), exceed minimum thresholds $r_{1_{\text{min}}}^{\text{X}}$, then the component Σ_1 will consume a given amount of said resources ($r_1^{\text{X}} = r_{1_{\text{min}}}^{\text{X}}$). We can use this viewpoint to analytically detect when our system may not function due to starvation from a given resource. This viewpoint also models the amount of loading each subsystem causes to other subsystems. We anticipate diverse future directions stemming from the research presented in this thesis as we build towards model-guided design for large-scale synthetic biological systems.

BIBLIOGRAPHY

[1] J.-Y. Trosset and P. Carbonell, "Synthetic biology for pharmaceutical drug discovery," *Drug design, development and therapy*, vol. 9, p. 6285, 2015.

[2] D. B. Pedrolli, N. V. Ribeiro, P. N. Squizato, V. N. de Jesus, D. A. Cozetto, *et al.*, "Engineering microbial living therapeutics: The synthetic biology toolbox," *Trends in biotechnology*, vol. 37, no. 1, pp. 100–115, 2019.

[3] L. N. Merk, A. S. Shur, A. Pandey, R. M. Murray, and L. N. Green, "Engineering logical inflammation sensing circuit for gut modulation," *bioRxiv*, pp. 2020–11, 2020.

[4] D. Nevozhay, T. Zal, and G. Balázsi, "Transferring a synthetic gene circuit from yeast to mammalian cells," *Nature communications*, vol. 4, no. 1, p. 1451, 2013.

[5] S. Cardinale, M. P. Joachimiak, and A. P. Arkin, "Effects of genetic variation on the e. coli host-circuit interface," *Cell reports*, vol. 4, no. 2, pp. 231–237, 2013.

[6] A. Boo, T. Ellis, and G.-B. Stan, "Host-aware synthetic biology," *Current Opinion in Systems Biology*, vol. 14, pp. 66–72, 2019.

[7] R. D. McCardell, A. Pandey, and R. M. Murray, "Control of density and composition in an engineered two-member bacterial community," *bioRxiv*, p. 632 174, 2019. DOI: `https://doi.org/10.1101/632174`,

[8] B. A. Blount, T. Weenink, and T. Ellis, "Construction of synthetic regulatory networks in yeast," *FEBS Letters*, vol. 586, no. 15, pp. 2112–2121, 2012.

[9] W. C. Ruder, T. Lu, and J. J. Collins, "Synthetic biology moving into the clinic," *Science*, vol. 333, no. 6047, pp. 1248–1252, 2011.

[10] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, pp. 339–342, 2000.

[11] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, 2000.

[12] M. Itaya, K. Fujita, A. Kuroki, and K. Tsuge, "Bottom-up genome assembly using the bacillus subtilis genome vector," *Nature Methods*, vol. 5, no. 1, pp. 41–43, 2008.

[13] Z. A. Tuza, V. Singhal, J. Kim, and R. M. Murray, "An in silico modeling toolbox for rapid prototyping of circuits in a biomolecular "breadboard" system," in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 1404–1410.

[14] V. Stein and K. Alexandrov, “Protease-based synthetic sensing and signal amplification,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 45, pp. 15 934–15 939, 2014.

[15] Z. Kis, H. S. Pereira, T. Homma, R. M. Pedrigi, and R. Krams, “Mammalian synthetic biology: Emerging medical applications,” *Journal of The Royal Society Interface*, vol. 12, no. 106, p. 20 141 000, 2015.

[16] B. P. Landry and J. J. Tabor, “Engineering diagnostic and therapeutic gut bacteria,” *Bugs as Drugs: Therapeutic Microbes for the Prevention and Treatment of Disease*, pp. 331–361, 2018.

[17] *Build-a-cell consortium*. [Online]. Available: `https://www.buildacell.org/`.

[18] A. Quarteroni, “Mathematical models in science and engineering,” *Notices of the AMS*, vol. 56, no. 1, pp. 10–19, 2009.

[19] D. D. Holm, J. E. Marsden, and T. S. Ratiu, “Euler-Poincaré models of ideal fluids with nonlinear dispersion,” *Physical Review Letters*, vol. 80, no. 19, p. 4173, 1998.

[20] R. D. Jones, Y. Qian, V. Siciliano, B. DiAndreth, J. Huh, *et al.*, “An endoribonuclease-based feedforward controller for decoupling resource-limited genetic modules in mammalian cells,” *bioRxiv*, p. 867 028, 2020.

[21] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie, “Stochastic modelling of gene regulatory networks,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 15, no. 15, pp. 691–711, 2005.

[22] B. Munsky, B. Trinh, and M. Khammash, “Listening to the noise: Random fluctuations reveal gene network parameters,” *Molecular Systems Biology*, vol. 5, no. 1, p. 318, 2009.

[23] A. Boo, T. Ellis, and G.-B. Stan, “Host-aware synthetic biology,” *Current Opinion in Systems Biology*, vol. 14, pp. 66–72, 2019.

[24] J. Cello, A. V. Paul, and E. Wimmer, “Chemical synthesis of poliovirus cdna: Generation of infectious virus in the absence of natural template,” *Science*, vol. 297, no. 5583, pp. 1016–1018, 2002.

[25] P. Srinivasan and C. D. Smolke, “Biosynthesis of medicinal tropane alkaloids in yeast,” *Nature*, vol. 585, no. 7826, pp. 614–619, 2020.

[26] W. Poole, A. Pandey, A. Shur, Z. A. Tuza, and R. M. Murray, “BioCRNpyler: Compiling chemical reaction networks from biomolecular parts in diverse contexts,” *PLOS Computational Biology*, vol. 18, no. 4, e1009987, 2022. DOI: `https://doi.org/10.1371/journal.pcbi.1009987`,

[27] F. N. Santos Navarro, “Multi-scale host-aware modeling for analysis and tuning of synthetic gene circuits for bioproduction,” Ph.D. dissertation, Universitat Politècnica de València, 2022.

[28] C. J. Myers, N. Barker, K. Jones, H. Kuwahara, C. Madsen, *et al.*, “Ibiosim: A tool for the analysis and design of genetic circuits,” *Bioinformatics*, vol. 25, no. 21, pp. 2848–2849, 2009.

[29] K. Choi, J. K. Medley, M. König, K. Stocking, L. Smith, *et al.*, “Tellurium: An extensible Python-based modeling environment for systems and synthetic biology,” *Biosystems*, vol. 171, pp. 74–79, 2018.

[30] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, *et al.*, “Copasi — A complex pathway simulator,” *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 2006.

[31] A. Pandey*, W. Poole*, A. Swaminathan*, V. Hsiao, and R. M. Murray, “Fast and flexible simulation and parameter estimation for synthetic biology using bioscrape,” *Journal of Open Source Software*, vol. 8, no. 83, p. 5057, 2023. DOI: `https://doi.org/10.21105/joss.05057`,

[32] A. Pandey and R. M. Murray, “Model reduction tools for phenomenological modeling of input-controlled biological circuits,” *bioRxiv*, 2020. DOI: `10.1101/2020.02.15.950840`. [Online]. Available: `https://www.biorxiv.org/content/early/2020/02/20/2020.02.15.950840`.

[33] L. Pasotti, M. Bellato, D. De Marchi, and P. Magni, “Mechanistic models of inducible synthetic circuits for joint description of DNA copy number, regulatory protein level, and cell load,” *Processes*, vol. 7, no. 3, p. 119, 2019.

[34] M. K. Transtrum and P. Qiu, “Bridging mechanistic and phenomenological models of complex biological systems,” *PLoS Computational Biology*, vol. 12, no. 5, e1004915, 2016.

[35] P. Fuentes, F. Zhou, A. Erban, D. Karcher, J. Kopka, *et al.*, “A new synthetic biology approach allows transfer of an entire metabolic pathway from a medicinal plant to a biomass crop,” *elife*, vol. 5, e13664, 2016.

[36] C. Y. Hu and R. M. Murray, “Layered feedback control overcomes performance trade-off in synthetic biomolecular networks,” *Nature Communications*, vol. 13, no. 1, p. 5393, 2022.

[37] D. Siegal-Gaskins, Z. A. Tuza, J. Kim, V. Noireaux, and R. M. Murray, “Gene circuit performance characterization and resource usage in a cell-free “breadboard”,” *ACS Synthetic Biology*, vol. 3, no. 6, pp. 416–425, 2014.

[38] N. J. Gaut and K. P. Adamala, “Reconstituting natural cell elements in synthetic cells,” *Advanced Biology*, vol. 5, no. 3, p. 2 000 188, 2021.

[39] B. Lavickova and S. J. Maerkl, “A simple, robust, and low-cost method to produce the pure cell-free system,” *ACS Synthetic Biology*, vol. 8, no. 2, pp. 455–462, 2019.

[40] D. Del Vecchio and R. M. Murray, *Biomolecular Feedback Systems*. Princeton University Press Princeton, NJ, 2015.

[41] D. T. Gillespie, “Stochastic simulation of chemical kinetics,” *Annual Reviews Physical Chemistry*, vol. 58, pp. 35–55, 2007.

[42] S. MacNamara, K. Burrage, and R. B. Sidje, “Multiscale modeling of chemical kinetics via the master equation,” *Multiscale Modeling & Simulation*, vol. 6, no. 4, pp. 1146–1168, 2008.

[43] J. Gunawardena, “Chemical reaction network theory for in-silico biologists,” *Notes available for download at http://vcp. med. harvard. edu/papers/crnt. pdf*, 2003.

[44] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck, “Computation with finite stochastic chemical reaction networks,” *Natural Computing*, vol. 7, no. 4, pp. 615–633, 2008.

[45] T. Schmiedl and U. Seifert, “Stochastic thermodynamics of chemical reaction networks,” *The Journal of Chemical Physics*, vol. 126, no. 4, p. 044 101, 2007.

[46] F. P. Davidescu and S. B. Jørgensen, “Structural parameter identifiability analysis for dynamic reaction networks,” *Chemical Engineering Science*, vol. 63, no. 19, pp. 4754–4762, 2008.

[47] E. Cinquemani, “Identifiability and reconstruction of biochemical reaction networks from population snapshot data,” *Processes*, vol. 6, no. 9, p. 136, 2018.

[48] V. Hsiao, A. Swaminathan, and R. M. Murray, “Control theory for synthetic biology: Recent advances in system characterization, control design, and controller implementation for synthetic biology,” *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 32–62, 2018.

[49] C. Y. Hu, J. D. Varner, and J. B. Lucks, “Generating effective models and parameters for rna genetic circuits,” *ACS Synthetic Biology*, vol. 4, no. 8, pp. 914–926, 2015.

[50] A. M. Stuart, “Inverse problems: A Bayesian perspective,” *Acta Numerica*, vol. 19, pp. 451–559, 2010.

[51] E. Appleton, C. Madsen, N. Roehner, and D. Densmore, “Design automation in synthetic biology,” *Cold Spring Harbor Perspectives in Biology*, vol. 9, no. 4, a023978, 2017.

[52] A. Gyorgy and D. Del Vecchio, “Limitations and trade-offs in gene expression due to competition for shared cellular resources,” in *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 5431–5436.

[53] P. Kokotović, H. K. Khalil, and J. O’Reilly, *Singular perturbation methods in control: Analysis and design*. SIAM, 1999.

[54] S. Z. Rizvi, J. Mohammadpour, R. Tóth, and N. Meskin, “A kernel-based pca approach to model reduction of linear parameter-varying systems,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1883–1891, 2016.

[55] B. Moore, “Principal component analysis in linear systems: Controllability, observability, and model reduction,” *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.

[56] S. Rao, A. Van der Schaft, K. Van Eunen, B. M. Bakker, and B. Jayawardhana, “A model reduction method for biochemical reaction networks,” *BMC Systems Biology*, vol. 8, no. 1, pp. 1–17, 2014.

[57] A. Gorban, “Model reduction in chemical dynamics: Slow invariant manifolds, singular perturbations, thermodynamic estimates, and analysis of reaction graph,” *Current Opinion in Chemical Engineering*, vol. 21, pp. 48–59, 2018.

[58] P. Kokotovic and P. Sannuti, “Singular perturbation method for reducing the model order in optimal control design,” *IEEE Transactions on Automatic Control*, vol. 13, no. 4, pp. 377–384, 1968.

[59] R. Marino and S. Nicosia, “Singular perturbation techniques in the adaptive control of elastic robots,” *IFAC Proceedings Volumes*, vol. 18, no. 16, pp. 95–100, 1985.

[60] K. R. Schneider and T. Wilhelm, “Model reduction by extended quasi-steady-state approximation,” *Journal of Mathematical Biology*, vol. 40, no. 5, pp. 443–450, 2000.

[61] N. Vora and P. Daoutidis, “Nonlinear model reduction of chemical reaction systems,” *AIChE Journal*, vol. 47, no. 10, pp. 2320–2332, 2001.

[62] F. G. Helfferich, “Systematic approach to elucidation of multistep reaction networks,” *The Journal of Physical Chemistry*, vol. 93, no. 18, pp. 6676–6681, 1989.

[63] J. A. Christiansen, “The elucidation of reaction mechanisms by the method of intermediates in quasi-stationary concentrations,” in *Advances in Catalysis*, vol. 5, Elsevier, 1953, pp. 311–353.

[64] T. Turanyi, A. Tomlin, and M. Pilling, “On the error of the quasi-steady-state approximation,” *The Journal of Physical Chemistry*, vol. 97, no. 1, pp. 163–172, 1993.

[65] A. Yannacopoulos, A. Tomlin, J. Brindley, J. Merkin, and M. Pilling, “The error of the quasi steady-state approximation in spatially distributed systems,” *Chemical Physics Letters*, vol. 248, no. 1-2, pp. 63–70, 1996.

[66] A. Papachristodoulou, Y.-C. Chang, E. August, and J. Anderson, “Structured model reduction for dynamical networked systems,” in *49th IEEE Conference on Decision and Control (CDC)*, IEEE, 2010, pp. 2670–2675.

[67] R. Babaghasabha, M. A. Khosravi, and H. D. Taghirad, “Adaptive robust control of fully constrained cable robots: Singular perturbation approach,” *Nonlinear Dynamics*, vol. 85, no. 1, pp. 607–620, 2016.

[68] L. A. Belhaj, M. Ait-Ahmed, and M. F. Benkhoris, “Embarked electrical network robust control based on singular perturbation model,” *ISA Transactions*, vol. 53, no. 4, pp. 1143–1151, 2014.

[69] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Dover, 2009.

[70] M. K. Fan, A. L. Tits, and J. C. Doyle, “Robustness in the presence of joint parametric uncertainty and unmodeled dynamics,” in *1988 American Control Conference*, IEEE, 1988, pp. 1195–1200.

[71] J. Freudenberg, D. Looze, and J. Cruz, “Robustness analysis using singular value sensitivities,” *International Journal of Control*, vol. 35, no. 1, pp. 95–116, 1982.

[72] X. Yang, J. J. Zhu, and A. S. Hodel, “Singular perturbation margin and generalised gain margin for linear time-invariant systems,” *International Journal of Control*, vol. 88, no. 1, pp. 11–29, 2015.

[73] X. Yang and J. J. Zhu, “Singular perturbation margin and generalised gain margin for nonlinear time-invariant systems,” *International Journal of Control*, vol. 89, no. 3, pp. 451–468, 2016.

[74] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 2002, vol. 3.

[75] J. Schmidt, *G. dahlquist, stability and error bounds in the numerical integration of ordinary differential equations. 85 S. Stockholm 1959. K. Tekniska Högskolans Handlingar*, 1961.

[76] T. Ström, “On logarithmic norms,” *SIAM Journal on Numerical Analysis*, vol. 12, no. 5, pp. 741–753, 1975.

[77] R. F. Snider, “Perturbation variation methods for a quantum Boltzmann equation,” *Journal of Mathematical Physics*, vol. 5, no. 11, pp. 1580–1587, 1964.

[78] R. M. Wilcox, “Exponential operators and parameter differentiation in quantum physics,” *Journal of Mathematical Physics*, vol. 8, no. 4, pp. 962–982, 1967.

[79] H. Tsai, K. Chan, *et al.*, “A note on parameter differentiation of matrix exponentials, with applications to continuous-time modelling,” *Bernoulli*, vol. 9, no. 5, pp. 895–919, 2003.

[80] R. P. Dickinson and R. J. Gelinas, “Sensitivity analysis of ordinary differential equation systems - A direct method,” *Journal of Computational Physics*, vol. 21, no. 2, pp. 123–143, 1976.

[81] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Springer Science & Business Media, 2006.

[82] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: A Lyapunov-based approach*. Princeton University Press, 2011.

[83] V. Chellaboina, W. M. Haddad, D. S. Bernstein, and D. A. Wilson, "Induced convolution operator norms of linear dynamical systems," *Mathematics of Control, Signals and Systems*, vol. 13, no. 3, pp. 216–239, 2000.

[84] V. Zahedzadeh, H. J. Marquez, and T. Chen, "Upper bounds for induced operator norms of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1159–1165, 2009.

[85] T. T. Georgiou, "On the computation of the gap metric," *Systems & Control Letters*, vol. 11, no. 4, pp. 253–257, 1988.

[86] B. D. Anderson, T. S. Brinsmead, and F. De Bruyne, "The vinnicombe metric for nonlinear operators," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1450–1465, 2002.

[87] V. Zahedzadeh, H. J. Marquez, and T. Chen, "On the computation of an upper bound on the gap metric for a class of nonlinear systems," in *2008 American Control Conference*, IEEE, 2008, pp. 1917–1922.

[88] A. Pandey, "Auto-reduce : Python based automated model reduction package," [Online].

[89] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, *et al.*, "Sympy: Symbolic computing in python," *PeerJ Computer Science*, vol. 3, e103, 2017.

[90] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, *et al.*, "The systems biology markup language (sbml): A medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[91] T. J. Snowden, P. H. Van Der Graaf, and M. J. Tindall, "A combined model reduction algorithm for controlled biochemical systems," *BMC Systems Biology*, vol. 11, no. 1, p. 17, 2017.

[92] X. Yang and J. J. Zhu, "Singular perturbation margin assessment of linear time-invariant systems via the Bauer-Fike theorems," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 6521–6528.

[93] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[94] A. N. Tikhonov, "Systems of differential equations containing small parameters in the derivatives," *Matematicheskii Sbornik*, vol. 73, no. 3, pp. 575–586, 1952.

[95] A. Goeke, S. Walcher, and E. Zerz, "Determining "small parameters" for quasi-steady state," *Journal of Differential Equations*, vol. 259, no. 3, pp. 1149–1180, 2015.

[96] T. J. Snowden, P. H. van der Graaf, and M. J. Tindall, "Methods of model reduction for large-scale biological systems: A survey of current methods and trends," *Bulletin of Mathematical Biology*, vol. 79, no. 7, pp. 1449–1486, 2017.

[97] M. Gasparyan, A. Van Messem, and S. Rao, "An automated model reduction method for biochemical reaction networks," *Symmetry*, vol. 12, no. 8, p. 1321, 2020.

[98] O. Buse, R. Pérez, and A. Kuznetsov, "Dynamical properties of the repressilator model," *Physical Review E*, vol. 81, no. 6, p. 066 206, 2010.

[99] C. Y. Hu and R. M. Murray, "Design of a genetic layered feedback controller in synthetic biological circuitry," *bioRxiv*, p. 647 057, 2019.

[100] A. J. Meyer, T. H. Segall-Shapiro, E. Glassey, J. Zhang, and C. A. Voigt, "Escherichia coli "marionette" strains with 12 highly optimized small-molecule sensors," *Nature Chemical Biology*, vol. 15, no. 2, pp. 196–204, 2019.

[101] A. Pandey and R. M. Murray, "A two-state ribosome and protein model can robustly capture the chemical reaction dynamics of gene expression," *bioRxiv*, 2020.

[102] X. Ren, A.-A. Baetica, A. Swaminathan, and R. M. Murray, "Population regulation in microbial consortia using dual feedback control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 5341–5347.

[103] M. Galdzicki, K. P. Clancy, E. Oberortner, M. Pocock, J. Y. Quinn, *et al.*, "The synthetic biology open language (sbol) provides a community standard for communicating designs in synthetic biology," *Nature Biotechnology*, vol. 32, no. 6, pp. 545–550, 2014.

[104] T. L. La Fleur, A. Hossain, and H. M. Salis, "Automated model-predictive design of synthetic promoters to control transcriptional profiles in bacteria," *bioRxiv*, 2021.

[105] H. M. Salis, "The ribosome binding site calculator," in *Methods in Enzymology*, vol. 498, Elsevier, 2011, pp. 19–42.

[106] T. Radivojević, Z. Costello, K. Workman, and H. Garcia Martin, "A machine learning automated recommendation tool for synthetic biology," *Nature Communications*, vol. 11, no. 1, pp. 1–14, 2020.

[107] G. Naseri and M. A. Koffas, "Application of combinatorial optimization strategies in synthetic biology," *Nature Communications*, vol. 11, no. 1, pp. 1–14, 2020.

[108] J. A. McLaughlin, C. J. Myers, Z. Zundel, G. Mısırlı, M. Zhang, *et al.*, "Synbiohub: A standards-enabled design repository for synthetic biology," *ACS Synthetic Biology*, vol. 7, no. 2, pp. 682–688, 2018.

[109] J. Hérisson, T. Duigou, M. Du Lac, K. Bazi-Kabbaj, M. S. Azad, *et al.*, "Galaxy-synbiocad: Automated pipeline for synthetic biology design and engineering," *Nature Communications*, 2022.

[110] S. Konur, L. Mierla, H. Fellermann, C. Ladroue, B. Brown, *et al.*, "Toward full-stack in silico synthetic biology: Integrating model specification, simulation, verification, and biological compilation," *ACS Synthetic Biology*, vol. 10, no. 8, pp. 1931–1945, 2021.

[111] D. Bryce, R. P. Goldman, M. DeHaven, J. Beal, B. Bartley, *et al.*, "Round trip: An automated pipeline for experimental design, execution, and analysis," *ACS Synthetic Biology*, vol. 11, no. 2, pp. 608–622, 2022.

[112] A. Pandey and R. M. Murray, "Robustness guarantees for structured model reduction of dynamical systems with applications to biomolecular models," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 9, pp. 5058–5086, 2023. DOI: `https://doi.org/10.1002/rnc.6013`,

[113] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, "Emcee: The mcmc hammer," *Publications of the Astronomical Society of the Pacific*, vol. 125, no. 925, p. 306, 2013.

[114] K. Rutherford, P. Yuan, K. Perry, R. Sharp, and G. D. Van Duyne, "Attachment site recognition and regulation of directionality by the serine integrases," *Nucleic Acids Research*, vol. 41, no. 17, pp. 8341–8356, 2013.

[115] A. C. Groth and M. P. Calos, "Phage integrases: Biology and applications," *Journal of Molecular Biology*, vol. 335, no. 3, pp. 667–678, 2004.

[116] M. C. Smith, R. Till, and M. C. Smith, "Switching the polarity of a bacteriophage integration system," *Molecular Microbiology*, vol. 51, no. 6, pp. 1719–1728, 2004.

[117] V. Hsiao, Y. Hori, P. W. Rothemund, and R. M. Murray, "A population-based temporal logic gate for timing and recording chemical events," *Molecular Systems Biology*, vol. 12, no. 5, p. 869, 2016.

[118] A. E. Friedland, T. K. Lu, X. Wang, D. Shi, G. Church, *et al.*, "Synthetic gene networks that count," *Science*, vol. 324, no. 5931, pp. 1199–1202, 2009.

[119] P. Siuti, J. Yazbek, and T. K. Lu, "Synthetic circuits integrating logic and memory in living cells," *Nature Biotechnology*, vol. 31, no. 5, pp. 448–452, 2013.

[120] A. I. Kim, P. Ghosh, M. A. Aaron, L. A. Bibb, S. Jain, *et al.*, "Mycobacteriophage bxb1 integrates into the mycobacterium smegmatis groel1 gene," *Molecular Microbiology*, vol. 50, no. 2, pp. 463–473, 2003.

[121] P. Ghosh, A. I. Kim, and G. F. Hatfull, "The orientation of mycobacteriophage bxb1 integration is solely dependent on the central dinucleotide of attp and attb," *Molecular Cell*, vol. 12, no. 5, pp. 1101–1111, 2003.

[122] P. Ghosh, L. R. Wasil, and G. F. Hatfull, "Control of phage bxb1 excision by a novel recombination directionality factor," *PLoS Biology*, vol. 4, no. 6, e186, 2006.

[123] F. P. Davidescu and S. B. Jørgensen, "Structural parameter identifiability analysis for dynamic reaction networks," *Chemical Engineering Science*, vol. 63, no. 19, pp. 4754–4762, 2008.

[124] A. Pandey, *Source code for integrase-excisionase characterization*. [Online]. Available: `https://github.com/ayush9pandey/integrase-excisionase-characterization`.

[125] C. A. Merrick, J. Zhao, and S. J. Rosser, "Serine integrases: Advancing synthetic biology," *ACS Synthetic Biology*, vol. 7, no. 2, pp. 299–310, 2018.

[126] J. Gunawardena, "Chemical reaction network theory for in-silico biologists," *Notes available for download at* `http://vcp.med.harvard.edu/papers/crnt.pdf`, vol. 5, 2003.

[127] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[128] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 03, pp. 90–95, 2007.

[129] *Bioscrape*. [Online]. Available: `https://github.com/biocircuits/bioscrape/`.

[130] I. Incer, A. Badithela, J. Graebener, P. Mallozzi, A. Pandey, *et al.*, "Pacti: Scaling assume-guarantee reasoning for system analysis and design," *arXiv*, p. 2303.17751, 2023. DOI: `https://doi.org/10.48550/arXiv.2303.17751`,

[131] M. H. Hanewich-Hollatz, Z. Chen, L. M. Hochrein, J. Huang, and N. A. Pierce, "Conditional guide rnas: Programmable conditional regulation of crispr/cas function in bacterial and mammalian cells via dynamic rna nanotechnology," *ACS central science*, vol. 5, no. 7, pp. 1241–1249, 2019.

[132] V. Hsiao, A. Swaminathan, and R. M. Murray, "Control theory for synthetic biology: Recent advances in system characterization, control design, and controller implementation for synthetic biology," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 32–62, 2018.

[133] J. A. Stapleton, K. Endo, Y. Fujita, K. Hayashi, M. Takinoue, *et al.*, "Feedback control of protein expression in mammalian cells by tunable synthetic translational inhibition," *ACS Synthetic Biology*, vol. 1, no. 3, pp. 83–88, 2012.

[134] M. Koch, J.-L. Faulon, and O. Borkowski, “Models for cell-free synthetic biology: Make prototyping easier, better, and faster,” *Frontiers in Bioengineering and Biotechnology*, p. 182, 2018.

[135] C. M. Kasey, M. Zerrad, Y. Li, T. A. Cropp, and G. J. Williams, “Development of transcription factor-based designer macrolide biosensors for metabolic engineering and synthetic biology,” *ACS Synthetic Biology*, vol. 7, no. 1, pp. 227–239, 2018.

[136] M. Cvijovic, J. Almquist, J. Hagmar, S. Hohmann, H.-M. Kaltenbach, *et al.*, “Bridging the gaps in systems biology,” *Molecular Genetics and Genomics*, vol. 289, no. 5, pp. 727–734, 2014.

[137] D. K. Agrawal, E. M. Dolan, N. E. Hernandez, K. M. Blacklock, S. D. Khare, *et al.*, “Mathematical models of protease-based enzymatic biosensors,” *ACS Synthetic Biology*, vol. 9, no. 2, pp. 198–208, 2020.

[138] T. Matsu-Ura, A. A. Dovzhenok, S. T. Coradetti, K. R. Subramanian, D. R. Meyer, *et al.*, “Synthetic gene network with positive feedback loop amplifies cellulase gene expression in neurospora crassa,” *ACS Synthetic Biology*, vol. 7, no. 5, pp. 1395–1405, 2018.

[139] P. Guerra, L.-A. Vuillemenot, B. Rae, V. Ladyhina, and A. Milias-Argeitis, “Systematic in vivo characterization of fluorescent protein maturation in budding yeast,” *ACS Synthetic Biology*, 2022.

[140] D. Del Vecchio, Y. Qian, R. M. Murray, and E. D. Sontag, “Future systems and control research in synthetic biology,” *Annual Reviews in Control*, vol. 45, pp. 5–17, 2018.

[141] O. Buse, R. Pérez, and A. Kuznetsov, “Dynamical properties of the repressilator model,” *Physical Review E*, vol. 81, no. 6, p. 066 206, 2010.

[142] C. D. McBride and D. Del Vecchio, “Predicting composition of genetic circuits with resource competition: Demand and sensitivity,” *ACS Synthetic Biology*, vol. 10, no. 12, pp. 3330–3342, 2021.

[143] J. T. Rapp, B. J. Bremer, and P. A. Romero, “Self-driving laboratories to autonomously navigate the protein fitness landscape,” *bioRxiv*, pp. 2023–05, 2023.

[144] S. M. Oliveira and D. Densmore, “Hardware, software, and wetware codesign environment for synthetic biology,” *BioDesign Research*, vol. 2022, 2022.

[145] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, “Taming Dr. Frankenstein: Contract-based design for cyber-physical systems*,” *European Journal of Control*, vol. 18, no. 3, pp. 217–238, 2012, ISSN: 0947-3580. DOI: `https://doi.org/10.3166/ejc.18.217-238`.

[146] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, *et al.*, "A contract-based methodology for aircraft electric power system design," *IEEE Access*, vol. 2, pp. 1–25, 2014. DOI: `10.1109/ACCESS.2013.2295764`.

[147] A. Saoud, A. Girard, and L. Fribourg, "Assume-guarantee contracts for continuous-time systems," *Automatica*, vol. 134, p. 109 910, 2021, ISSN: 0005-1098. DOI: `https://doi.org/10.1016/j.automatica.2021.109910`.

[148] I. Filippidis and R. M. Murray, "Layering assume-guarantee contracts for hierarchical system design," *Proceedings of the IEEE*, vol. 106, no. 9, pp. 1616–1654, 2018.

[149] A. Sangiovanni-Vincentelli and G. Martin, "Platform-based design and software design methodology for embedded systems," *IEEE Design & Test of computers*, vol. 18, no. 6, pp. 23–33, 2001.

[150] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, *et al.*, "Contracts for system design," *Foundations and Trends®in Electronic Design Automation*, vol. 12, no. 2-3, pp. 124–400, 2018, ISSN: 1551-3939.

[151] I. Incer, A. L. Sangiovanni-Vincentelli, C.-W. Lin, and E. Kang, "Quotient for assume-guarantee contracts," in *16th ACM-IEEE International Conference on Formal Methods and Models for System Design*, ser. MEMOCODE'18, Beijing, China, Oct. 2018, pp. 67–77, ISBN: 9781538661956. DOI: `10.1109/MEMCOD.2018.8556872`.

[152] R. Passerone, I. Incer, and A. L. Sangiovanni-Vincentelli, "Coherent extension, composition, and merging operators in contract models for system design," *ACM Transactions on Embedded Computer Systems*, vol. 18, no. 5s, Oct. 2019, ISSN: 1539-9087. DOI: `10.1145/3358216`.

[153] I. Incer, *The Algebra of Contracts*. University of California, Berkeley, 2022.

[154] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone, *et al.*, "Multiple viewpoint contract-based specification and design," in *International Symposium on Formal Methods for Components and Objects*, Springer, 2007, pp. 200–225.

[155] J. C. Anderson, C. A. Voigt, and A. P. Arkin, "Environmental signal integration by a modular and gate," *Molecular Systems Biology*, vol. 3, no. 1, p. 133, 2007.

[156] S. Jayanthi, K. S. Nilgiriwala, and D. Del Vecchio, "Retroactivity controls the temporal dynamics of gene transcription," *ACS Synthetic Biology*, vol. 2, no. 8, pp. 431–441, 2013.

[157] A. Gyorgy and D. Del Vecchio, "Modular composition of gene transcription networks," *PLoS Computational Biology*, vol. 10, no. 3, e1003486, 2014.

[158] L. M. Hochrein, H. Li, and N. A. Pierce, “High-performance allosteric conditional guide rnas for mammalian cell-selective regulation of CRISPR/Cas,” *ACS Synthetic Biology*, vol. 10, no. 5, pp. 964–971, 2021.

[159] A. Gyorgy, “A practical step-by-step guide for quantifying retroactivity in gene networks,” in *Synthetic Gene Circuits*, Springer, 2021, pp. 293–311.

[160] T. Moriya, T. Yamaoka, Y. Wakayama, S. Ayukawa, Z. Zhang, *et al.*, “Comparison between effects of retroactivity and resource competition upon change in downstream reporter genes of synthetic genetic circuits,” *Life*, vol. 9, no. 1, p. 30, 2019.