Engineering the Feedback Dynamics of *in vitro* Synthetic Biological Systems

Joshua D. Bishop

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2012

Reading Committee:
Eric Klavins, Chair
Herbert Sauro
Georg Seelig

Program Authorized to Offer Degree: Electrical Engineering

University of Washington

Abstract

Engineering the Feedback Dynamics of *in vitro* Synthetic Biological Systems

Joshua D. Bishop

Chair of the Supervisory Committee:
Associate Professor Eric Klavins
Electrical Engineering

Building a biochemical system from scratch that rivals a living cell in ability to robustly perform many complex tasks in response to environmental signals is beyond the state-of-the-art. However, feedback is clearly a principle widely employed by cells – as it is by engineers – to enable robust dynamical behaviors, although a limited system-level understanding of feedback regulation and dynamical behavior in biochemical contexts hampers our ability to engineer such systems. Studying simple feedback systems in a rich, yet fully synthetic, biochemical context – like that provided by DNA nanotechnology – may therefore lead to a new state-of-the-art for synthetic biological systems.

This thesis, in keeping with this philosophy, describes basic efforts in engineering biochemical feedback control systems in the form of simple, *in vitro*, nucleic-acid-based devices. I describe two such devices, which provide basic test-beds for engineering feedback dynamics *in vitro*. The first device is a DNA nanomotor, previously described in the literature, that is built from and operated by nucleic acid components, and that I modify by the introduction of a protein enzyme to improve the performance of the device in experimental tests. The second device I design and build from nucleic acid components and two protein enzymes to regulate the free quantity of an RNA molecule, which in turn can be used to dynamically drive the operation of other nucleic-acid-based systems –such as the first device – in a robust manner.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Eight years in graduate school is a long time, and so I have many people to thank. First and foremost, I thank my thesis advisor, Dr. Eric Klavins, who has led me through a wide-ranging exploration of mathematics, engineering and science. I began graduate school with fuzzy ideas about researching computing elements at the nanoscale, and after disappointment and subsequent detour into Robotics and Control Engineering, my nascent research vision was rescued when I met Eric. He then stuck with me during the long years of slow progress towards completing my thesis in the bio-lab, for which I am eternally grateful (even though it was his idea to work with DNA in the first place!). I have too many thank you's for Eric to list here.

I also thank everyone who has gone before me in Eric's research group. I like to think that the research of Drs. JM McNew, Nils Napp, and Faw Shaw was easier because they stuck with robotics, but in fact they are all top-quality researchers whose works were an inspiration to me. The group has also had several wonderful post-doctoral researchers with whom it was a pleasure to collaborate. Thus I thank Drs. David Thorsley, Kyle Havens, and especially Danny Georgiev, whose early work on modeling the genelet systems forms the bedrock on which Chapter 5 of this thesis was built.

Thanks also to the other graduate students in Eric's group: Kevin Oishi, for his enthusiasm for research, especially that related to *in vitro* DNA technology, and his willingness to discuss problems anytime, anywhere. Rob Egbert, whose time was coveted by many, including myself, for his mastery in the lab, his clear vision of and facility with high-level concepts, and his easy-going nature. Shelly Jang, for extensive discussions of modeling and data-fitting, and her open nature. Chris Takahashi, for both trips through the department, his deep knowledge of control theory, and his technical mastery.

I also want to thank everyone who made like in the new bio-lab better and more in-

teresting: Nick Bolton and Tammy Gu; the members of the Seelig group (especially Yuan Chen), the UW iGEM teams, and the participants in the Intro to Synthetic Biology lab classes.

I have also had many, many interesting and often useful interactions with other researchers in the field. First I would like to thank my reading committee members, Georg Seelig and Herbert Sauro, without which this thesis would literally not be possible. Georg provided important reagents to me in the early stages of the genelet regulation project, as well as countless hours of advice and moral support. Herbert provided the use of his lab's fluorescence plate reader to me in the months leading up to my defense, when our lab's reader was non-functioning, and I desperately needed results. Discussions with Herbert on the nature of my research problems, and his perspective on the intersection of synthetic biology and control theory have opened the door to future collaborations.

Outside of UW, I am hugely grateful to the support of the Molecular Programming Project and its members. I am unbelievably lucky to have had the opportunity to discuss my research with, and receive feedback from, these leaders in the field. I especially want to acknowledge the contributions of Drs. Jongmin Kim, Elisa Franco, and Erik Winfree to this thesis, whose feedback and insight into my problems broke down barriers it is unlikely I would have been able to push through by myself.

Finally, I want to thank all my friends and family, who have (mostly) patiently waited all this time to call me "Doctor."

DEDICATION

to my dear future wife, Gretchen, and to my family, without all of whom this document would have never been written.

Chapter 1

INTRODUCTION

Cells are the fundamental units of living organisms. They exist in great variety and complexity and are capable of performing a dizzying array of tasks, often in highly uncertain environments. The robust ability of cells to live and self-replicate in uncertain environments is due in large part to the extensive biochemical regulatory systems that control cell state in response to environmental inputs. Feedback, which underpins the complex, dynamical structure of these systems, is widely studied in biology and engineering due to its ability to shape important characteristics of dynamical behavior. However, in cells, these dynamical behaviors are evolved properties and therefore resistant to a rational understanding of the design principles, feedback or otherwise, that underlie them.

Engineered biochemical regulatory systems, on the other hand, might promise, in similar contexts, the same robust, controllable behavior as the regulatory systems in cells, but with the advantage of having been designed with an inherent understanding to a particular purpose. A broader goal of my research is to move forward our ability to engineer feedback-regulated biochemical systems that approach the complexity, scale, and robust performance of natural biological systems, and to provide tools that allow the orthogonal investigation of the regulatory principles underlying natural biological systems. For example, take as context the field of DNA nanotechnology, which has leveraged the simple mechanism of Watson-Crick base-pairing to implement an impressively wide array of behaviors, including self-assembly, amplification, detection, computation, sensing, and actuation. Taking a cell as a model, layering the appropriate regulatory behaviors on top of these others might enable the construction of even more robust and/or complex behaviors.

Certainly, building a biochemical system from scratch that rivals a cell in ability to robustly perform many complex tasks in response to environmental signals is beyond the state-of-the-art. However, feedback is clearly a principle widely employed by cells – as it is

by engineers – to enable robust dynamical behaviors, although a limited system-level understanding of feedback regulation and dynamical behavior in biochemical contexts hampers our ability to engineer such systems. Studying simple feedback systems in a rich, yet fully synthetic, biochemical context – like that provided by DNA nanotechnology – may therefore lead to a new state-of-the-art for synthetic biological systems.

This thesis, in keeping with this philosophy, describes basic efforts in engineering biochemical feedback control systems in the form of simple, *in vitro*, nucleic-acid-based devices. I describe two such devices, which provide basic test-beds for engineering feedback dynamics *in vitro*. The first device is a DNA nanomotor, previously described in the literature, that is built from and operated by nucleic acid components, and that I modify by the introduction of a protein enzyme to improve the performance of the device in experimental tests. The second device I design and build from nucleic acid components and two protein enzymes to regulate the free quantity of an RNA molecule, which in turn can be used to dynamically drive the operation of other nucleic-acid-based systems –such as the first device – in a robust manner.

In Chapter 2, I survey results in synthetic biology and DNA nanotechnology that provide an engineering context to this work. I also discuss the general study of feedback regulation in biological contexts.

In Chapter 3, I discuss work on improving a DNA nanomotor to make it less sensitive to waste products in experimental tests. Guided by chemical reaction network models of the nanomotor system, two successive modifications are made to the nanomotor system that compensate for waste, and analysis and experiments are presented that confirm the modifications improve the nanomotor's performance.

In Chapter 4, I introduce my work on using feedback to regulate the supply of RNA *fuel*, such as might operate the DNA nanomotor discussed in Chapter 3. In this chapter, I introduce the design of two *genelet* systems, unregulated and regulated RNA delivery devices, that we use to compare our ability to regulate the RNA fuel supply. The process used to design these systems is discussed in detail, from high-level regulatory network specifications to low-level sequence specifications.

In Chapter 5, I introduce a set of models that we use to analyze the behavior of the

genelet systems introduced in Chapter 4. The models predict qualitative features for the response of both systems regarding dynamic equilibria, tunability, and robustness to disturbances in a fuel-consuming, downstream process.

Chapter 6, I present experimental tests of the unregulated and regulated genelet systems. The tests demonstrate the impact of tuning in both genelet systems, highlight the relatively robust nature of correctly tuned regulated genelet systems with respect to a fuel-consuming, downstream process, and generally validate the qualitative predictions made by the model analyses. The results of the application of several data-fitting methods are also presented, to draw a quantitative connection between models and experiment.

Appendices A, B, and C provide supplementary information for Chapters 4, 5, and 6, respectively.

Chapter 2

BACKGROUND

A driving force for the synthetic biology research community is the idea that one can expose the basic "design principles" of life by trying to build living systems from scratch. This idea holds for natural and synthetic constructs, in contexts ranging from isolated molecular circuits to single- or multi-cellular organisms, and across physical and theoretical models of living systems. For example, cellular automata are theoretical models for which self-replicating versions were first designed over 50 years ago [75]. Since then cellular automata have provided models for the study and simulation of biological systems [56]; one even bears the name "Game of Life" [7].

The most common physical model for living systems, however, is the single cell, which in many respects seems to be the fundamental unit of life [65]. Consequently, one approach to understanding living systems is to build an artificial cell from the "bottom-up," completely from scratch [26]. Such an effort requires complex, dynamical i/o systems implemented by interacting nano-scale particles, but does not necessarily require nucleic acids, proteins, lipids, organelles, or other recognizable, natural cellular constituents. On the other hand, natural components provide an obvious template for "bottom-up" efforts.

Thus, a first approximation to a completely artificial cell is the synthetic cell, which is a "top-down" effort to provide a minimally living platform for "bottom-up" biological engineering. Researchers have developed genetic recombination techniques that allow for the construction of entire genomes from chemically synthesized DNA [31]. The technique was used to recreate the genome of the bacterium *Mycoplasma mycoides* from synthetic parts, which was then recombined into the yeast *Saccharomyces cerevisiae* and transplanted into the genome-restricted shell of the related bacterium *Mycoplasma capricolum* [32]. Eventually, however, creating a platform for general biological engineering will require a synthetic genome containing a minimal set of genes sufficient for life [33, 60, 34], rather than a syn-

thetic recreation of a naturally occurring genome.

With such a platform not yet available, most synthetic biologists are in practice restricted to building engineered biological *circuits* inside existing organisms. This practice may in fact be desirable, however, if the goal is to engineer new behaviors in those particular organisms. The idea to engineer new behaviors in living organisms has been around since at least 1961, when Jacob and Monod noted of bacterial genetic regulatory systems: "it is obvious from the analysis of these mechanisms that their known elements could be connected into a wide variety of 'circuits' endowed with any desired degree of stability" [43]. This observation has eventually led to the engineering of *genetic* circuits, where the basic elements – genes, RNA, proteins, and other biochemical components – are rationally designed to perform specific interactions and are often borrowed from other organisms.

Widespread practice in engineering genetic circuits lagged considerably behind the recognition of its potential, however. In 1995, McAdams and Shapiro, noting the similarity in the dynamic functions of genetic networks and electrical circuits, applied electrical engineering circuit analysis techniques to characterize the natural genetic circuit used by the λ bacteriophage to choose between lysis and lysogeny [82] in terms of signal timing, feedback, time delays, and sequential logic [66]. McAdams was therefore well-poised to review two of the first demonstrated examples of synthetic genetic circuits: the now-canonical Repressilator [24] and genetic toggle switch [30], which appeared in the same issue of the journal *Nature* in 2000. These two studies also employed models to qualitatively describe the dynamic behavior of genetic circuits, but the circuits were designed and implemented specifically to exhibit this behavior (oscillations and bistability, respectively) *in vivo*.

Hasty *et al.* also reviewed these, and several more, early examples of engineered genetic circuits in 2002, claiming that the construction of these circuits was finally enabled by advances in nonlinear dynamics, computing power, and genome-level sequencing. "[T]he gene[tic] circuit approach," they wrote, "uses mathematical and computational tools in the analysis of a proposed circuit diagram, while novel experimental techniques are used to construct the networks according to the model blueprint" [39]. The authors further underlined the similarity of the engineered genetic circuit approach to engineering methodology by casting the circuits as genetic *modules* [67, 38, 57, 44] that may be studied as simplified

systems, or interconnected into more complex systems that enable greater levels of cellular control.

However, "despite tremendous individual successes in genetic engineering and biotechnology," such as those discussed in the previous reviews, "why is the engineering of useful synthetic biological systems still an expensive, unreliable and ad hoc research process?" asked Endy in 2005 [25]. Endy posited two answers: 1) biology is too complex and/or poorly understood to engineer, and 2) no foundational technologies exist to make engineering biology an engineering problem.

The first concern is a non-issue since, according to Endy, "these concerns are best evaluated by attempting to surmount them" [25]. This is a position also taken by Benner and Sismour, who argued, in their 2005 review of the field, that "synthesis" can drive discovery in a way that pure observation and analysis cannot [6]. Also that same year, Sprinzak and Elowitz reviewed the field with a focus on "how synthetic biology can address biological questions at the level of genetic circuits" [105]. The uniting theme in these commentaries is, once again, the goal of using synthetic biology to expose the basic "design principles" of life.

The second concern is an issue for which "one practical starting point," according to Endy, "is to consider past lessons from when other engineering disciplines emerged from the natural sciences" [25]. This perspective was followed closely in 2006 by Andrianantoandro *et al.*, who examined the application of engineering rules to synthetic biology using specific examples [2]. Similarly, but from a more commercial viewpoint, Baker *et al.* in 2006 envisioned a "Bio Fab" in which "principles and practices learned from engineering successes can help transform biotechnology from a specialized craft into a mature industry" [3]. More recently, Carlson supported the eponymous argument of his 2010 book Biology is Technology by drawing an analogy between current biological engineering and the early aeronautics industry [11]. All of these works outline three main ideas that together form a more developed conceptual framework for engineering biology: *standardization*, *decoupling*, and *abstraction*.

Standardization should be a familiar concept to anyone who has used a screwdriver, browsed the Internet, or tried to plug a North American hairdryer into a European outlet. Standards are needed to ensure that technologies are safe, compatible, interoperable, and of

high quality. The difficulties of developing standards for engineering genetic circuits occur at many levels. The behavior of an individual genetic part is affected by its embedding in a particular circuit, organism, growth environment, and possibly even a particular laboratory. Therefore, a strong basis for standardization is the development of platforms that allow researchers to define, characterize, share, and apply parts in a variety of contexts.

One such platform is the Registry of Standard Biological Parts [1], which was initially built on an early synthetic biological technical standard known as the BioBrick [98]. The BioBrick standard defines specific methods for encoding, combining, and propagating genetic circuitry in and between laboratories. The success of the Registry, and the associated BioBrick standard, has been largely driven by the International Genetically Engineered Machines (iGEM) competition, in which teams of undergraduates from around the world both use and contribute biological parts to the Registry while competing to build the "best" genetic circuit. The competition, due to its format, serves to foster an open exchange of ideas and knowledge in synthetic biology between researchers, students, and the general public.

Public opinion of synthetic biology is in many ways tied to its potential applications. While the upsides of the field are many, and have been discussed often, safety in synthetic biology is an incredibly important topic due to a perceived potential for environmental contamination, public health epidemics, and bioterrorism. This topic was recently addressed in the United States by a Presidential Commission for the Study of Bioethical Issues, which released a 2010 report urging "monitoring and dialogue between the private and public sectors to achieve open communication and cooperation" [37]. The commission backed this moderate recommendation with two observations: 1) organisms have tended to evolve away synthetic functionalities, and 2) synthetic organisms are still difficult to build, as I have already noted. Nonetheless, standards will also help to support the open communication and cooperation urged by the Commission.

Decoupling refers to the separation of a difficult problem – such as engineering genetic circuits – into easier sub-problems, the solutions to which add up to a solution of the original problem. One example of decoupling in engineering is the separation of design and fabrication. The Registry of Standard Biological Parts and advances in DNA synthesis and assembly, respectively, are (at least partially) the results of an effort to achieve this

separation in biological engineering. Another example of decoupling in engineering is modularity, where functional modules with well-defined interfaces represent solutions to easier sub-problems, and the predictable behavior arising from their careful interconnection represents the solution to a more difficult problem.

Abstraction provides a conceptual basis for the process of assembling a complex genetic circuit from a collection of relatively simple modules, which are in turn assembled from a collection of relatively simple biochemical parts. An abstraction of an object contains only the information about that object that is useful to a particular purpose. For the purpose of designing a large, complicated circuit, for example, a useful abstraction of a module might contain only a definition of its functionality and interfaces. The design process in engineering often uses a hierarchy of abstraction layers, where the details of implementation at a given layer are hidden to the layers at a higher level of abstraction.

An oft-cited example of an abstraction hierarchy in engineering is one used to design computer networks: physical parts, gates, modules, computers, and networks, from low- to high-level. Andrianantoandro *et al.* suggested this example is analogous to a biological hierarchy: biomolecules, biochemical reactions, pathways, cells, and tissues or cultures, from low- to high-level [2]. Endy suggested a different hierarchy to use in biological engineering: DNA, parts, devices, and systems [25]. Marguet *et al.* elaborated on how the latter abstraction hierarchy might guide the application of engineering tools, in a 2005 review on biology by design, highlighting the interplay of DNA synthesis-based fabrication with the DNA layer, rational design with the parts layer, computational modeling with the circuits layer, and monitoring technologies with the systems layer.

Fundamentally, abstraction hierarchies are useful ways to manage complexity. Complexity is limited across the hierarchy because higher levels of abstraction ignore implementation details while lower levels of abstraction decouple systems into separate modules. The cost of this structure, however, is the need to ensure compatibility between component modules at the interfaces between layers, an issue typically addressed through the use of standardized parts.

Heinemann and Panke provided a similar summary of this emerging engineering vision of synthetic biology – with a caveat – when they wrote in 2006: "Building novel parts,

devices and in particular complex systems will require a systematic approach that relies on modularity and abstraction at various cellular levels in order to be useful... [and] a technology to fabricate the designs. This is synonymous to following an engineering approach, even if biology is in many aspects not understood well enough to consider it a sufficient knowledge base for an engineering discipline" [41]. Two such poorly-understood aspects of biology, discussed by Voigt in a review of the field written the same year, are 1) the diffusive nature of biochemical reaction systems, which physically limits both the decoupling and reuse of genetic parts, and 2) the living (evolving) nature of cells, which temporally limits the performance and permanence of genetic parts and their interconnections [110].

Nonetheless, synthetic biology researchers had by this time collectively demonstrated a "toolbox" of genetic modules whose robust performance in different designs led Voigt to catalog them as evidence that an engineering approach is still useful [110]. Three years later, Purnick and Weiss called the modules in this toolbox the "first wave of synthetic biology," in which "basic elements – for example promoters, ribosome binding sites and transcriptional repressors – were combined to form small modules with specified behaviours... [like] switches, cascades, pulse generators, time-delayed circuits, oscillators, spatial patterning and logic formulas" [84].

The first wave of synthetic biology, according to Purnick and Weiss, exposed three genetic circuit design principles that were used to build the modules in this "toolbox": 1) circuit optimization by directed evolution, 2) circuit selection from a library of circuit variants, and 3) circuit construction by rational iterative design [84]. (I will focus on the third design principle – rational iterative design – in this thesis, since it most reflects the engineering – and the other two the biological – aspects of synthetic biology.) The process of rational iterative design starts with a *specification*, an explicit set of requirements for the engineered system. The development and analysis of computational models of the system, followed by fabrication and experimental testing, then lead to refinements of the design. In this process a design is refined until the physical system meets the specification.

Purnick and Weiss, however, also wrote that "we are now on the cusp of the second wave of synthetic biology, in which basic parts and modules need to be integrated to create systems-level circuits," and that to accomplish this task, "synthetic biologists will formulate

new and effective bioengineering design principles" [84]. On the other hand, Mukherji and van Oudenaarden, in a review of the field written the same year,"highlight[ed] how the process of engineering biological systems – from synthetic promoters to the control of cell–cell interactions – has contributed to our understanding of how endogenous systems are put together and function" [73]. Together, these commentaries expose the complementary nature of the pairing of biology and engineering in synthetic biology, in which new design principles are discovered in pursuit of existing design principles and vice versa. As we try to build more robust, complicated, and useful synthetic systems, the better we understand living systems; the better we understand living systems, the easier it becomes to build more robust, complicated, and useful synthetic systems.

Smolke and Silver examined this exact synergy between synthetic and living systems in a 2011 essay, although they separated the study of synthetic and living systems into the fields of synthetic and systems biology, respectively. They wrote, in the conclusion of their essay:

> "Although each field could in principle exist without the other, we instead feel that the natural interplay between design, analysis, and understanding highlights the important relationship between systems biology and synthetic biology. Systems biology brings added layers of information that will further empower future efforts to design synthetic biological systems. Synthetic biology brings new technologies and tools that can be applied to effectively test our understanding of natural biological systems. By integrating the contributions of these rapidly evolving fields, scientists and engineers together will be well positioned to transform health, well-being, and the environment in the years to come" [102].

Systems-level thinking has certainly started to influence the direction of synthetic biology research and the most recent reviews of the field reflect this fact. Khalil and Collins, for example, reviewed the practical applications of synthetic biology in 2010, when "10 years after the demonstration of synthetic biology's inaugural devices, engineered biomolecular networks are beginning to move into the application stage and yield solutions to many complex societal problems" [47]. Nandagopal and Elowitz, in 2011, observed that "researchers

have begun to develop a new generation of synthetic circuits that integrate more closely with endogenous cellular processes... [and] are providing fundamental insights into the regulatory architecture, dynamics, and evolution of genetic circuits and enabling new levels of control across diverse biological systems" [74].

The growing capability of synthetic biologists to deal with complexity is evident not just in the increasing complexity of engineered biological systems, but also in the increasing complexity of the biological context. The reviews and commentaries referenced above mostly provide examples of biological engineering in bacterial cells. However, biological engineering is also increasingly being performed in yeast [53] and mammalian [111] cells, and human therapies are on the horizon [91].

For all the advances toward a better understanding of living systems made by synthetic biologists working *in vivo*, however, more fundamental approaches can be taken by decreasing the complexity of the biological context, i.e. building the components of an artificial cell *in vitro*. As Forster and Church remarked in 2007, "until we can assemble a form of life in vitro from defined, functionally understood macromolecules and small-molecule substrates, how can we say that we understand the secret of life?" [27]. Thus we return to the concept of "bottom-up" biological engineering with a discussion of *in vitro* synthetic biology.

The first cell was presumably constructed "bottom-up" by natural processes. The protocell represents this hypothetical "missing link" between the primordial soup and modern cellular organisms, consisting of a minimal set of self-replicating macromolecules [59] in a self-assembled membrane [9]. Significant efforts have been put towards the creation of a protocell in the laboratory in an attempt to recreate nature's path to the construction of complex living systems [72, 103, 23].

However, the construction of an artificial cell that rivals the complexity of natural living systems requires also building all of the fundamental systems that make up a living cell, whatever they may be, not just a minimal set of self-replicating macromolecules in a self-assembled membrane. Natural systems have, in the last decade, yielded yet another "top-down" approach in the form of cell-free extracts, which ease the need for completely synthetic mechanisms in an artificial cell. In particular, transcription/translation extracts, which are now commercially available from several sources (e.g. Ambion, Austin, TX;

New England Biolabs, Ipswich, MA), have been used to engineer cell-free activation and repression cascades [78] as well as green fluorescent protein (GFP) production inside in a phospholipid vesicle [79, 77]. These effort could be considered a first approximation to a totally artificial cell, yet still more basic approaches exist.

Advances in DNA nanotechnology [95] have enabled the construction of molecular computation systems at the nanoscale, which compute functions or perform basic tasks [113, 85, 54]. Theoretical work by Soloveichik has shown that DNA is a highly programmable molecule, a "universal substrate" for the implementation of arbitrary chemical kinetics [104]. In addition, Oishi and Klavins gave blueprints for the *in vitro* construction of linear I/O systems with DNA [80], suggesting that the large body of engineering practice associated with linear systems might be brought to bear in the context of an artificial cell. Thus, DNA nanotechnology might provide an artificial basis for building the fundamental systems required by an artificial cell capable of complex tasks.

Indeed, at the laboratory bench, nucleic acids alone have been used to implement a wide variety of *in vitro*, enzyme-free systems such as logic gates [94, 85], catalytic and feedback circuits [121, 122], self-assembling systems [90, 89, 115], triggered amplification circuits [21], and a host of autonomous nanomachines [46, 4] (also see [120] for a thorough review of this literature). All or some of these systems may provide crucial to building an artificial cell. For example, autonomous DNA nanomachines are devices that convert chemical energy to mechanical actuation and are typified by the DNA nanomotor [117, 62, 58, 101, 12, 16, 13, 70, 45] and the closely-related DNA walker [97, 99, 107, 116, 81, 35, 61]. In an artificial cell, nanomachines might act as synthetic replacements for protein-based molecular motors such as kinesin, playing a role in transport, enzymatic catalysis, and other important cellular processes.

The biological engineering abstraction hierarchy presented by Endy – DNA, parts, devices, and systems – is an informative guide to the construction of these *in vitro*, enzyme-free, DNA-based devices. At the lowest level there is the DNA, where systems are implemented choosing the *sequences* (A's, C's, G's, and T's) of DNA *strands*. At the next level, there are the parts, of which the two fundamental types are DNA hybridization and toehold-mediated DNA strand displacement. At higher levels, there are a wide array of devices and systems

that can be constructed by clever combination of just these two parts, as demonstrated by the references above.

Hybridization parts are built from distinct, but complementary, single-stranded DNA (ssDNA) sequences. Complementary DNA bases (A and T, C and G) bind via individually weak hydrogen bonds, and longer complementary sequences bind collectively to form more strongly bound double-stranded DNA (dsDNA). Toehold-mediated strand displacement parts are built from two distinct, but complementary, sequences and a third sequence identical to one of the other two except shorter by several bases on one end. Initially, the shorter sequence is bound to its longer complement, which leaves an exposed *toehold*. The other long sequence then binds to the toehold, creating a branch point between the three co-localized sequences. The branch point moves in a random walk as the weakly bound bases to either side of the branch point unbind and bind randomly, and eventually the displacing sequence either completely displaces the shorter sequence or dissociates from the toehold. The length of the toehold determines the kinetics of this process in the test tube [123].

The canonical example of *in vitro*, enzyme-free systems is the DNA tweezer, described by Yurke *et al* – coincidentally – in the journal *Nature* in 2000, and built from a simple combination of these two part types [117]. As already noted, the larger systems referenced above use the same two parts originally demonstrated by Yurke *et al.*. Some design principles that have emerged from the further use of hybridization and strand-displacement parts, however, are the use of *domains*, which decouple functionally independent sequences in designs [15], to represent well-characterized hybridization [124, 19, 20, 18, 63, 64, 118, 119] and strand-displacement [123] parts; the development of chemical reaction network kinetics models [104, 80, 54] that predict the behavior of large systems, e.g. in [94, 86]; and the practice of orchestrating sequential events through either the covering or uncovering of toeholds, e.g. in [94, 86].

Another, broader, class of systems exists for engineering biochemical circuits *in vitro*: enzyme-dependent nucleic acid systems. In fact, this class of systems forms the backbone of molecular biology, encompassing PCR, restriction-based and other types of cloning, isothermal nucleic acid amplification and detection [76], and many common biochemical assays. Purified protein enzymes, as molecular components, are a counterpoint to synthesized nu-

cleic acids. They are highly evolved, highly specific, and (currently) difficult to design. However, they provide a toolbox of new parts that are difficult, if not impossible, to recreate with nucleic acids alone, and, luckily, many interface naturally with nucleic acids.

As it turns out, enzyme-dependent nucleic acid systems provide fertile ground for the exploration of *dynamic* biological behaviors *in vitro*. They fall between enzyme-free DNA and cell-extract systems in terms of complexity, which allows a rich space of behaviors to explore, while still providing a relatively simple biochemical environment for this exploration. Two synthetic frameworks were recently introduced that allow exploration in this space.

In 2007, Kim *et al.* introduced the transcriptional switch, or *genelet* [51], an *in vitro*, enzyme-dependent, nucleic-acid-based regulatory component. As an example of the flexibility of genelet-based systems, a bistable switch [51], several oscillatory topologies [52, 28], a transcriptional rate regulator [29], and a positive-feedback switch [106] have all been built and tested experimentally. Genelet systems combine hybridization and strand-displacement parts with two new enzyme parts: Ribonuclease H (R_H), which hydrolytically degrades RNA from an RNA/DNA hybrid complex [40, 69], and T7 RNA polymerase (R_P), which binds to dsDNA at a sequence-dependent location, and generates an RNA copy of the downstream dsDNA sequence.

In 2011, Montagne *et al.* introduced an alternative set of *in vitro*, enzyme-dependent, nucleic-acid-based regulatory components [71]. These systems rely on priming protected DNA templates for extension by a DNA polymerase enzyme. The newly extended primers are then released from the template by a nickase enzyme and feed forward or back to regulate production or repression from the same template or others. The framework is demonstrated through the construction of an oscillatory network, which should be noted for the thorough experimental characterization of every reaction involved in the network. However, these systems are run at temperatures close to the melting temperature of the DNA templates, which makes them non-ideal candidates for the regulation of other *in vitro* DNA systems.

Chapter 3

AN IMPROVED *IN VITRO* DNA NANOMOTOR

3.1 Introduction

There are many *in vitro* DNA systems and devices whose operation is driven or controlled by one or more nucleic acid *fuel* strands. Any such system or device is eligible for dynamic control through the regulation of fuel production, but are more often operated by large initial concentrations or repeated exogenous additions of fuel. The DNA nanomotor is a basic device that requires fuel to cycle through a series of operating states.

We consider the DNA nanomotor introduced by Chen, Wang, and Mao [12] a model system in this regard. This nanomotor is constructed from a 10-23 DNA enzyme [92] (DNAzyme), as shown in Fig. 3.1a and cycles autonomously as the DNAzyme binds to and cleaves nucleic acid fuel strands into waste strands. This specification implies that the fuel is completely digested by the nanomotor. Unfortunately, the fuel and waste strands have similar affinities for the nanomotor. The result is that the accumulation of waste significantly interferes with the operation of the system, which can be observed experimentally as a loss of performance, as shown in Fig. 3.1b, which represents our reproduction of the experiment by Chen, Wang, and Mao [12] (with minor modifications). We therefore iterate on the design of the nanomotor, using models to guide design refinements that improve its performance in experiments.

To begin, an explanation for the observed behavior of the DNAzyme nanomotor follows, accompanied by a descriptive mathematical model, which together suggest a modification of the DNAzyme nanomotor system. We call this modified system the compensated system, as it incorporates a waste management mechanism and shows improved performance in experiments. We also introduce a streamlined version of the compensated system that simplifies the design of the nanomotor and still maintains improved performance. Both the compensated and streamlined systems use the enzyme ribonuclease H [40, 69] (R_H) to digest

waste strands so that they do not interfere with the normal operation of the nanomotor. Finally, we offer an explanation, loosely based on the idea of disturbance rejection [22] from control theory, for the improved behavior of the compensated and streamlined systems. Specifically, we show that the original system is highly sensitive to the presence of waste, while the other systems are not.

This chapter contains material published in the journal article "An Improved Autonomous DNA Nanomotor" [8].

3.2 Design 1: Original System

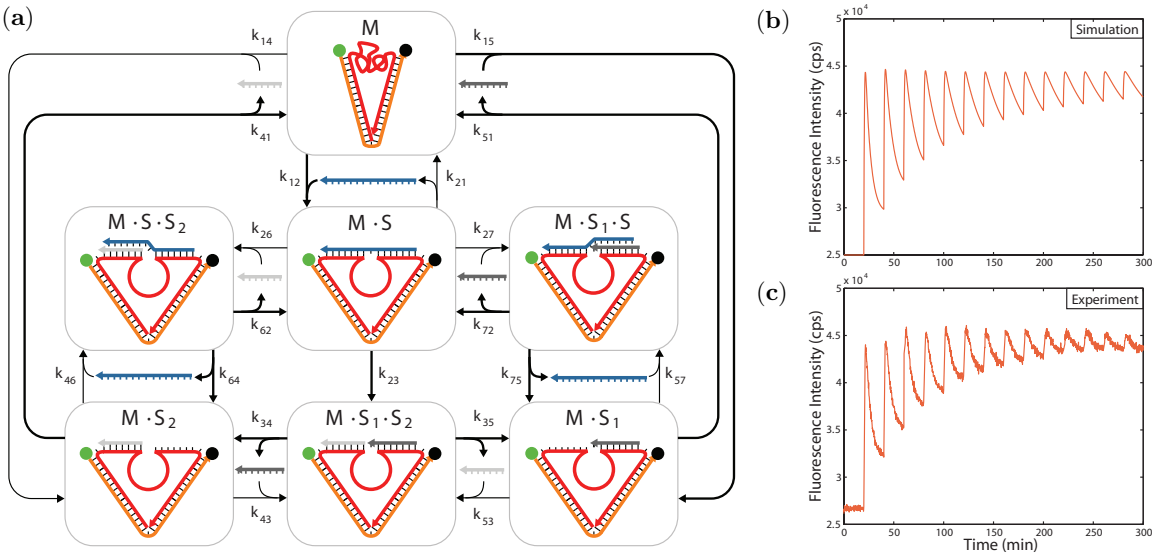

Figure 3.1: (**a**) The reaction network model of the original DNAzyme nanomotor system. The states of the nanomotor are boxed and labeled by species, where M denotes the nanomotor, S denotes the RNA fuel, and S_1 and S_2 denote the waste products. The symbol $(\cdot)$ denotes the hybridization of species into a larger complex. The model captures the operation of the nanomotor as well as the effect of waste having non-negligible affinity for the nanomotor. All reactions are reversible hybridization reactions, excepting the fuel-cleaving reaction, and the direction of these reactions is indicated by bold arrows. Parameters and further details are described in the supplementary documentation. The model yields a simulated trajectory (**b**) of the system that is in agreement with our experimental results (**c**). In these plots, spikes in fluorescence intensity correspond to the addition of a stoichiometric quantity of RNA fuel to the solution of nanomotors.

The DNAzyme nanomotor introduced by Chen, Wang, and Mao [12] consists of two helical arms joined by a single-base hinge on one end and an RNA-cleaving 10-23 DNAzyme on the other (see Fig. 3.1a). The DNAzyme region and an RNA fuel strand bind to form a bulged-duplex hybrid, or the *open* conformation of the nanomotor. The DNAzyme eventually cleaves the bound fuel strand and collapses, (hypothetically) causing the dissociation of the cleaved pieces of the fuel stand and the *closed* conformation of the nanomotor.

Each nanomotor is marked with a fluorophore/quencher pair at opposite ends of the DNAzyme, so that open or closed conformations of an ensemble of nanomotors can be indirectly observed via fluorescence resonance energy transfer (FRET) with a spectrofluorometer. The operation of the DNAzyme nanomotor is demonstrated in a FRET experiment in which stoichiometric quantities of the RNA fuel are periodically added to a solution of the nanomotor. The experimental procedure is described in the Materials and Methods portion of this chapter.

Ideally, when the RNA fuel is cleaved, the waste products are completely dissociated and inert. In this case, the addition of fuel in a FRET experiment would be marked by a rapid rise in fluorescence intensity, followed by an exponential decline to the *same* base fluorescence level observed before the addition, as the ensemble of nanomotors collectively used the entirety of the fuel. In other words, the specification for the expected behavior of the system upon continued, periodic additions of fuel is a periodic signal with constant amplitude. However, in experiments, the system does not return to the base fluorescence level after each addition of fuel, but instead saturates at a high level of fluorescence, as shown in Fig. 3.1c, which is our reproduction of the experiment by Chen, Wang, and Mao [12]. In our experiments, we use RNA fuel strands instead of RNA/DNA chimeric strands and a different fluorophore/quencher pair, yet the results are qualitatively similar to those of Chen, Wang, and Mao [12] in that the fluorescent output does not return to its basal level after each addition.

To give insight into why this behavior occurs, we describe a deterministic chemical reaction network (CRN) model of the general nanomotor system dynamics, which we write compactly as

$$\dot{\mathbf{v}} = \mathbf{A} \cdot \mathbf{K}(\mathbf{v}), \tag{3.1}$$

where the vector $\mathbf{v}$ holds the concentrations of relevant species, the matrix $\mathbf{A}$ holds the stoichiometries of the reactions, and the function $\mathbf{K}$ returns the propensities of each reaction. We choose a deterministic model due to the relatively large concentration of all the species in experiments. The same species are considered for all models, so that

$$\mathbf{v} = ([M], [M \cdot S], [M \cdot S_1 \cdot S_2], [M \cdot S_1], [M \cdot S_2], [M \cdot S_1 \cdot S], [M \cdot S \cdot S_2], [S], [S_1], [S_2])^T$$

in each model. Here, M denotes the nanomotor, S the RNA fuel, S_1 and S_2 the waste products, and the symbol $(\cdot)$ denotes the hybridization of these species in a larger complex. For example, $M \cdot S$ denotes the open nanomotor complex, where the fuel is bound to the DNAzyme in a bulged-duplex hybrid.

The specific model of the original system includes 20 reversible hybridization reactions and one irreversible cleaving reaction as shown graphically in Fig. 3.1a. Therefore, we describe the original system dynamics as in (3.1) with

$$\dot{\mathbf{v}} = \mathbf{A}_o \cdot \mathbf{K}_o(\mathbf{v}), \tag{3.2}$$

where

$$\mathbf{A}_o = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_o(\mathbf{v}) = \begin{pmatrix} k_{12}\, v_1(t)\, v_8(t) - k_{21}\, v_2(t) \\ k_{14}\, v_1(t)\, v_9(t) - k_{41}\, v_4(t) \\ k_{15}\, v_1(t)\, v_{10}(t) - k_{51}\, v_5(t) \\ k_{23}\, v_2(t) - k_{32}\, v_3(t) \\ k_{26}\, v_2(t)\, v_9(t) - k_{62}\, v_6(t) \\ k_{27}\, v_2(t)\, v_{10}(t) - k_{72}\, v_7(t) \\ k_{34}\, v_3(t) - k_{43}\, v_4(t)\, v_{10}(t) \\ k_{35}\, v_3(t) - k_{53}\, v_5(t)\, v_9(t) \\ k_{46}\, v_4(t)\, v_8(t) - k_{64}\, v_6(t) \\ k_{57}\, v_5(t)\, v_8(t) - k_{75}\, v_7(t) \end{pmatrix}.$$

However, none of the individual species described in this model are observable experimentally. Therefore, we define an output mapping

$$y(t) = (\alpha \cdot \mathbf{m}(t) + \beta), \tag{3.3}$$

where the vector $\mathbf{m}(t) = vol_{total} \cdot \mathbf{v}(t)$ contains the total mass of each species, in moles; the constant vector α contains the estimated fluorescence efficiency over the baseline level, per mass, in moles^{-1}, of each species; and β is the baseline fluorescence level, in photon counts per second (cps), for the given experimental conditions. In this definition, y is the total fluorescence output of the system, above the baseline, expressed in (cps).

The values in α are estimated using the expression for FRET efficiency

$$E = \frac{1}{1 + (\frac{r}{R_0})^6}, \tag{3.4}$$

with $R_0 = 3$ nm. The distance between fluorophore and quencher molecules, r, is estimated, for each species containing the nanomotor complex, to be the combined length of the attached fuel or waste products. These species are estimated to have a fluorescence contribution per mass proportional to their FRET efficiency. The closed nanomotor, fuel, and waste species are estimated to have zero fluorescence contribution above the base level. The resulting estimated fluorescence efficiencies from application of (3.4) are

$$\alpha = (0, 0.0105624, 0.0103512, 0.00392833, 0.00219512, 0.0103512, 0.0103512, 0, 0, 0).$$

The qualitative behavior does not depend on exact values, so no effort has been made to make these values precise.

We generate deterministic simulations of system output by numerically solving the model differential equations and applying the output mapping. Numerical integration of model equations is done separately for each period of fuel addition. The initial conditions of each period, starting immediately after an addition of fuel at time t, are

$$\mathbf{v}(t^+) = \frac{vol_{added} \cdot \mathbf{v}_{added} + vol(t^-) \cdot \mathbf{v}(t^-)}{vol_{added} + vol(t^-)},$$

which updates the final conditions of the previous period, $\mathbf{v}(t^-)$, to account for exogenous input, $\mathbf{v}_{added}$, and the resulting dilution $vol(t^-)/(vol(t^-) + vol_{added})$. Simulation data for each period are then appended to create a single output trajectory. An example simulation trajectory for the original system is shown in Fig. 3.1b, and it is in good agreement with the experimental data shown in Fig. 3.1b.

As the experimental and simulation data show, the actual behavior of the original system is a periodic signal with decreasing amplitude. The rate parameters we use to generate this and other simulations in this section are listed in Table 3.1, but we can prove that the model shown in Fig. 3.1a predicts this behavior no matter how the values of the rate parameters are chosen. Specifically, the model suggests that an increasingly large fraction of motors remain open after each addition of fuel due to the interference of the accumulating waste strands S_1 and S_2.

Table 3.1: Rate constant estimates used in the simulations shown in Figs. 3.1b, 3.2b, and 3.4b.

Model	Parameters					
original	k_{12}	1.0	(/M/s)	k_{21}	0.10	(/s)
	k_{14}	0.010	(/M/s)	k_{41}	1.0	(/s)
	k_{15}	0.10	(/M/s)	k_{51}	0.10	(/s)
	k_{23}	0.95	(/s)			
	k_{27}	0.10	(/M/s)	k_{72}	100	(/s)
	k_{34}	10.	(/s)	k_{43}	0.010	(/M/s)
	k_{35}	10.	(/s)	k_{53}	0.010	(/M/s)
	k_{46}	0.010	(/M/s)	k_{64}	10.	(/s)
	k_{57}	0.10	(/M/s)	k_{75}	100	(/s)
compensated	k_i	(same as orig. model)				
	k_E	0.83	(/s)			
streamlined	k_{12}	1.0	(/M/s)	k_{21}	0.10	(/s)
	k_E	0.83	(/s)			

The proof goes as follows. First, we consider the steady-state fluorescence output of the original system above its basal level of fluorescence by taking the limit of (3.3)

$$y^* = \lim_{t\to\infty} \left(\alpha^T \cdot \mathbf{m} + \beta\right) - \beta,$$

which can be rewritten as

$$y^* = vol_{total} \left(\alpha^T \cdot \mathbf{v}^*\right), \tag{3.5}$$

where $\mathbf{v}^*$ is the species concentration vector at steady-state.

Second, we assume that $y^* = 0$. Since α_1, α_8, α_9, and α_{10} are zero and all other values of α are positive, we have from the assumption $y^* = 0$ and the definition of y^* that $v_i^* = 0$, where $i \in \{2, 7\}$. In steady state, $\mathbf{A}_o \cdot \mathbf{K}_o(\mathbf{v}^*) = 0$, and substituting in the values of $\mathbf{v}^*$ that must be zero yields the equation

$$\begin{pmatrix} -v_1^* \left(k_{12}\, v_8^* + k_{14}\, v_9^* + k_{15}\, v_{10}^*\right) \\ k_{12}\, v_1^*\, v_8^* \\ 0 \\ k_{14}\, v_1^*\, v_9^* \\ k_{15}\, v_1^*\, v_{10}^* \\ 0 \\ 0 \\ -k_{12}\, v_1^*\, v_8^* \\ -k_{14}\, v_1^*\, v_9^* \\ -k_{15}\, v_1^*\, v_{10}^* \end{pmatrix} = 0.$$

For positive values of the rate parameters k_{12}, k_{14}, and k_{15}, this equation allows only two solutions: $v_1^* = 0$, or v_8^*, v_9^*, and $v_{10}^* = 0$. However, since $v_i^* = 0$, where $i \in \{2, 7\}$, these solutions imply that the system either contains no nanomotors, or no waste/fuel strands, respectively. Therefore, if fuel/waste strands and nanomotors are present in the original nanomotor system, y^* cannot equal zero.

3.3 Design 2: The Compensated System

We infer from the model of the original system that the waste strands compete with the fuel. One way to selectively digest waste is with the enzyme Ribonuclease H (R_H), which

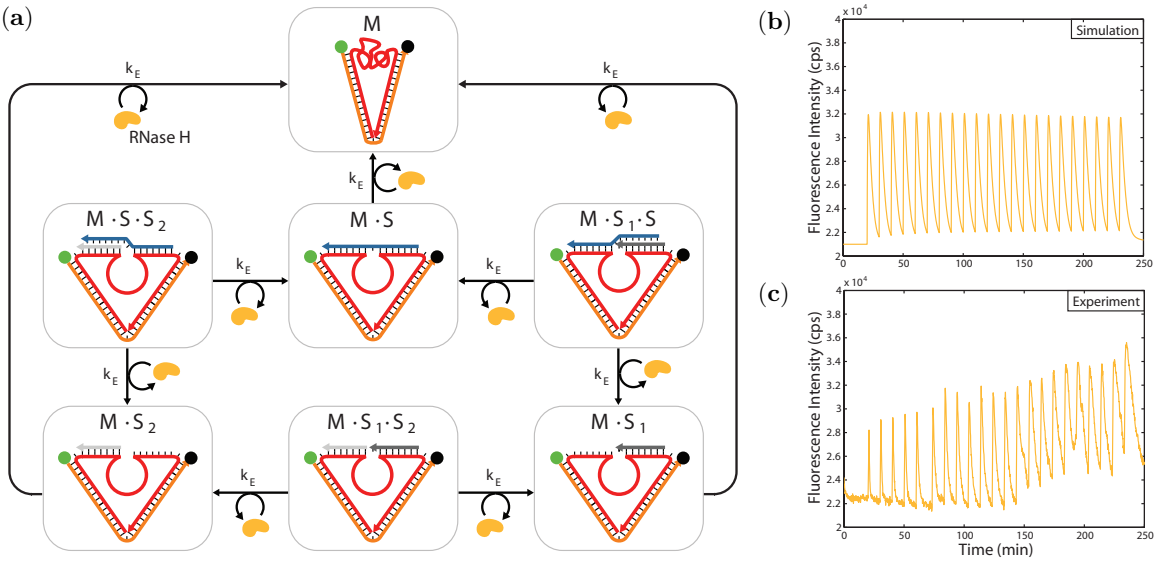

Figure 3.2: (**a**) The reaction network model of the R_H compensator. The compensated system model is the composition of these reactions and those in the original system model depicted in Fig. 3.1a. The enzyme R_H degrades fuel and waste bound to motors, mitigating the effect of accumulating waste on nanomotor performance. The model yields a simulated trajectory (**b**) of the compensated system that is in agreement with experimental results (**c**). The experimental data show an upward trend not reflected in the simulated trajectory, which may be a result of non-uniform mixing or RNAase H degradation, neither of which are modeled.

digests RNA into tri-ribonucleotide or smaller fragments, but only from RNA/DNA hybrids [40, 69]. This property of R_H ensures that the fuel is not digested before it binds to the nanomotor and that the waste products are degraded to the point of having negligible affinity for the nanomotor. In this section, we refine the design of the original nanomotor to include R_H, and evaluate a corresponding model and experimentally test the new system for improved performance.

The inclusion of R_H adds reaction pathways to the original system that are shown in Fig. 3.2a, with the estimated rate of RNA digestion by R_H listed in Table 3.1. We use a first-order model of enzyme kinetics for the compensated system model, so that we describe the compensated system dynamics as in (3.1) with

$$\dot{\mathbf{v}} = \mathbf{A}_c \cdot \mathbf{K}_c(\mathbf{v}), \tag{3.6}$$

where

$$\mathbf{A}_c^T = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}_c(\mathbf{v}) = \begin{pmatrix} k_{12}\, v_1(t)\, v_8(t) - k_{21}\, v_2(t) \\ k_{14}\, v_1(t)\, v_9(t) - k_{41}\, v_4(t) \\ k_{15}\, v_1(t)\, v_{10}(t) - k_{51}\, v_5(t) \\ k_{23}\, v_2(t) \\ k_{26}\, v_2(t)\, v_9(t) - k_{62}\, v_6(t) \\ k_{27}\, v_2(t)\, v_{10}(t) - k_{72}\, v_7(t) \\ k_{34}\, v_3(t) - k_{43}\, v_4(t)\, v_{10}(t) \\ k_{35}\, v_3(t) - k_{53}\, v_5(t)\, v_9(t) \\ k_{46}\, v_4(t)\, v_8(t) - k_{64}\, v_6(t) \\ k_{57}\, v_5(t)\, v_8(t) - k_{75}\, v_7(t) \\ k_E\, v_2(t) \\ k_E\, v_6(t) \\ k_E\, v_6(t) \\ k_E\, v_7(t) \\ k_E\, v_7(t) \\ k_E\, v_3(t) \\ k_E\, v_3(t) \\ k_E\, v_4(t) \\ k_E\, v_5(t) \end{pmatrix}.$$

Here, k_E is the first-order rate constant representing enzyme activity, which was estimated from experimental data. In particular, the initial slope of the decrease in fluorescence signal after the first addition of a stoichiometric quantity of fuel to a solution of nanomotors is considered an acceptable approximation to this rate constant.

The enzymatic degradation pathways mitigate the effect of the unintended pathways that dominate in the original system when the concentration of waste becomes significant. A

simulation of the resulting model is shown in Fig. 3.2b, which predicts that the performance of the compensated nanomotor system is improved over that of the original system.

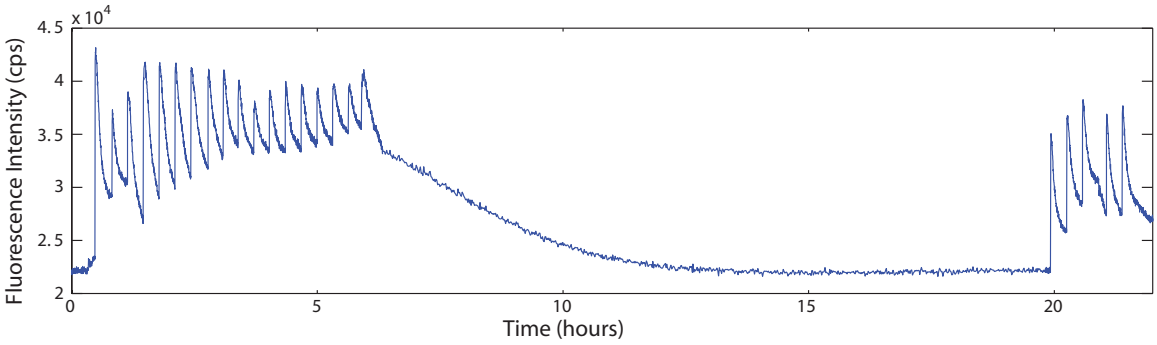

Figure 3.3: Experimental data demonstrating performance recovery of the compensated system after 14 hours of inactivity. In this experiment, only 20 units of R_H were added to the system, which may account for the loss in performance observed between adjacent periods.

Experiments confirm the behavior predicted by the modified model. Fig. 3.2c shows experimental data for repeated additions of fuel strands to the compensated system. Further experiments, as described in Fig. 3.3 show that the compensated nanomotor can be cycled even after 14 hours of inactivity.

One issue with this design, however, is that R_H does not distinguish between fuel bound to the nanomotor and waste bound to the nanomotor. In fact, it competes with the DNAzyme to digest fuel and therefore two enzymes are actually redundant.

3.4 Design 3: The Streamlined System

We introduce a streamlined design for the nanomotor in which the DNAzyme is replaced by a sequence of DNA that is exactly complementary to the RNA fuel as shown in Fig. 3.4a. This sequence binds with an RNA fuel strand into a full-duplex hybrid, opening the nanomotor. The R_H then digests the bound RNA strand, closing the nanomotor. As in the compensated system, autonomy is preserved without performance loss from the accumulation of waste, but with a simpler, non-redundant design.

To model the streamlined system, we consider all conformations in which RNA is bound

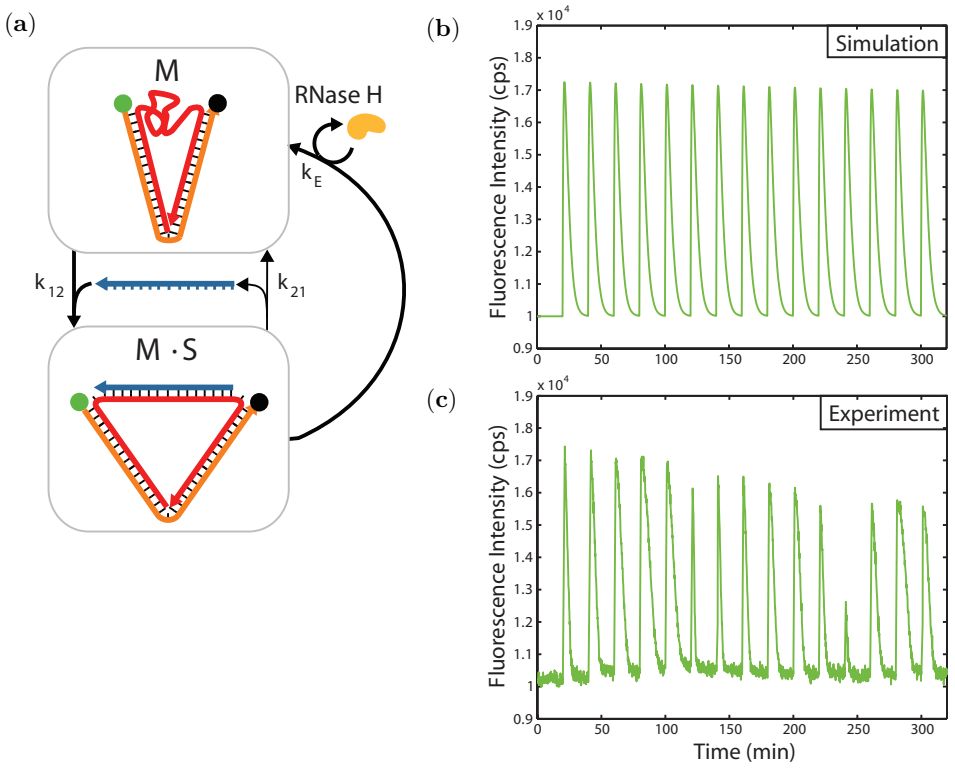

Figure 3.4: (**a**) The reaction network model of the streamlined system. The simplified nanomotor design yields a model with only two nanomotor states: *open* and *closed*. The motor opens when it binds with a fuel strand, and closes when the bound fuel is digested by R_H. The model yields a simulated trajectory (**b**) of the streamlined system that is in good agreement with experimental results (**c**). In experiments, R_H is replenished with every fifth addition of fuel to counteract the natural decay of the enzyme at 22^{o}C.

to the nanomotor as open (see Fig. 3.4a). We describe the streamlined system dynamics, much as before in (3.1), with

$$\dot{\mathbf{v}} = \mathbf{A}_s \cdot \mathbf{K}_s(\mathbf{v}), \tag{3.7}$$

where

$$\mathbf{A}_s = \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \mathbf{K}_s(\mathbf{v}) = \begin{pmatrix} k_{12}v_1(t)v_8(t) - k_{21}v_2(t) \\ k_E v_2(t) \end{pmatrix}.$$

A simulation of the resulting model is shown in Fig. 3.4b, which predicts the same improved performance seen in the compensated system.

Experimental data from the streamlined system confirm the behavior predicted by the model. In Fig. 3.4c we show experimental data for repeated additions of fuel to the streamlined system. In these experiments, the activity of R_H slowly decreases, possibly due to relatively long exposure to room temperature. Thus, we replenish the R_H with every fifth addition of fuel to the system.

3.5 Sensitivity Analysis

Biochemical nanodevices have been demonstrated in simple laboratory settings. However, to be useful, nanodevices must eventually be deployed in more complex settings and in the presence of other devices. Thus, a truly useful nanodevice must be robust to its operating environment. It is instructive to characterize each nanomotor system in the context of its robustness.

For example, given a model of the ideal behavior of the DNAzyme nanomotor, we can characterize the effect of increasing concentrations of waste products as a *disturbance*. For each system, a disturbance model is used that divides the system into $\mathbf{A} \cdot \hat{\mathbf{K}}(\mathbf{v})$, which contains the terms describing the ideal behavior of the system, and $\mathbf{A} \cdot \tilde{\mathbf{K}}(\mathbf{v})$, which contains

the terms describing the disturbance: competition from waste strands. In particular, we write the dynamics as

$$\dot{\mathbf{v}} = \mathbf{A} \cdot \hat{\mathbf{K}}(\mathbf{v}) + \varepsilon \mathbf{A} \cdot \tilde{\mathbf{K}}(\mathbf{v}). \tag{3.8}$$

where the stoichiometric matric $\mathbf{A}$ with the rate function $\hat{\mathbf{K}}$ models the ideal behavior (with no competition from the waste strands), and $\mathbf{A}$ with $\tilde{\mathbf{K}}$ models the disturbance (in which competition is present). We use the equilibrium fluorescence intensity

$$y^* = \lim_{t \to \infty} \left(\alpha^T \cdot \mathbf{m}(t) + \beta \right) - \beta,$$

as a performance metric.

In the disturbance model (3.8) description of the original system, the system dynamics are

$$\dot{\mathbf{v}} = \mathbf{A_o} \cdot \hat{\mathbf{K}}_o(\mathbf{v}) + \varepsilon \mathbf{A_o} \cdot \tilde{\mathbf{K}}_o(\mathbf{v}), \tag{3.9}$$

where $\mathbf{A}_o$ is as before,

$$\hat{\mathbf{K}}_o(\mathbf{v}) = \begin{pmatrix} k_{12}\, v_1(t)\, v_8(t) - k_{21}\, v_2(t) \\ -k_{41}\, v_4(t) \\ -k_{51}\, v_5(t) \\ k_{23}\, v_2(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \text{ and } \tilde{\mathbf{K}}_o(\mathbf{v}) = \begin{pmatrix} 0 \\ k_{14}\, v_1(t)\, v_9(t) \\ k_{15}\, v_1(t)\, v_{10}(t) \\ 0 \\ k_{26}\, v_2(t)\, v_9(t) - k_{62}\, v_6(t) \\ k_{27}\, v_2(t)\, v_{10}(t) - k_{72}\, v_7(t) \\ k_{34}\, v_3(t) - k_{43}\, v_4(t)\, v_{10}(t) \\ k_{35}\, v_3(t) - k_{53}\, v_5(t)\, v_9(t) \\ k_{46}\, v_4(t)\, v_8(t) - k_{64}\, v_6(t) \\ k_{57}\, v_5(t)\, v_8(t) - k_{75}\, v_7(t) \end{pmatrix}.$$

Likewise, in the disturbance model (3.8) description of the compensated system, the system dynamics are

$$\dot{\mathbf{v}} = \mathbf{A_c} \cdot \hat{\mathbf{K}}_c(\mathbf{v}) + \varepsilon \mathbf{A_c} \cdot \tilde{\mathbf{K}}_c(\mathbf{v}), \tag{3.10}$$

where $\mathbf{A}_c$ is as before,

$$\hat{\mathbf{K}}_c(\mathbf{v}) = \begin{pmatrix} k_{12}\, v_1(t)\, v_8(t) - k_{21}\, v_2(t) \\ -k_{41}\, v_4(t) \\ -k_{51}\, v_5(t) \\ k_{23}\, v_2(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ k_E\, v_2(t) \\ k_E\, v_6(t) \\ k_E\, v_6(t) \\ k_E\, v_7(t) \\ k_E\, v_7(t) \\ k_E\, v_3(t) \\ k_E\, v_3(t) \\ k_E\, v_4(t) \\ k_E\, v_5(t) \end{pmatrix}, \text{ and } \tilde{\mathbf{K}}_c(\mathbf{v}) = \begin{pmatrix} 0 \\ k_{14}\, v_1(t)\, v_9(t) \\ k_{15}\, v_1(t)\, v_{10}(t) \\ 0 \\ k_{26}\, v_2(t)\, v_9(t) - k_{62}\, v_6(t) \\ k_{27}\, v_2(t)\, v_{10}(t) - k_{72}\, v_7(t) \\ k_{34}\, v_3(t) - k_{43}\, v_4(t)\, v_{10}(t) \\ k_{35}\, v_3(t) - k_{53}\, v_5(t)\, v_9(t) \\ k_{46}\, v_4(t)\, v_8(t) - k_{64}\, v_6(t) \\ k_{57}\, v_5(t)\, v_8(t) - k_{75}\, v_7(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Finally, in the disturbance model (3.8) description of the streamlined system, the system dynamics are

$$\dot{\mathbf{v}} = \bar{\mathbf{A}}_s \cdot \hat{\mathbf{K}}_s(\mathbf{v}) + \varepsilon \bar{\mathbf{A}}_s \cdot \tilde{\mathbf{K}}_s(\mathbf{v}), \tag{3.11}$$

where

$$\bar{\mathbf{A}}_s = \begin{pmatrix} -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\hat{\mathbf{K}}_s(\mathbf{v}) = \begin{pmatrix} k_{12}\, v_1(t)\, v_8(t) - k_{21}\, v_2(t) \\ -k_{41}\, v_4(t) \\ -k_{51}\, v_5(t) \\ k_E v_2(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ k_E\, v_2(t) \\ k_E\, v_6(t) \\ k_E\, v_6(t) \\ k_E\, v_7(t) \\ k_E\, v_7(t) \\ k_E\, v_3(t) \\ k_E\, v_3(t) \\ k_E\, v_4(t) \\ k_E\, v_5(t) \end{pmatrix}, \text{ and } \tilde{\mathbf{K}}_s(\mathbf{v}) = \begin{pmatrix} 0 \\ k_{14}\, v_1(t)\, v_9(t) \\ k_{15}\, v_1(t)\, v_{10}(t) \\ 0 \\ k_{26}\, v_2(t)\, v_9(t) - k_{62}\, v_6(t) \\ k_{27}\, v_2(t)\, v_{10}(t) - k_{72}\, v_7(t) \\ k_{34}\, v_3(t) - k_{43}\, v_4(t)\, v_{10}(t) \\ k_{35}\, v_3(t) - k_{53}\, v_5(t)\, v_9(t) \\ k_{46}\, v_4(t)\, v_8(t) - k_{64}\, v_6(t) \\ k_{57}\, v_5(t)\, v_8(t) - k_{75}\, v_7(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The stoichiometric matrix of the non-disturbance model of the streamlined system, $\mathbf{A}_s$ in (3.7), lacks a description of the waste competition reactions, since we assume the fuel is completely digested by the enzyme R_H. For purposes of performing the comparative sensitivity analysis, the stoichiometric matrix requires the addition of these reactions, which is reflected in the matrix $\bar{\mathbf{A}}_s$ in (3.11).

The initial condition used in the analysis is $\mathbf{v}(0) = (0.5, 0, 0, 0, 0, 0, 0, 0.5, 0, 0)^T$. The variable ε ranges over the interval $(0, 1]$. In Fig. 3.5 we show the *sensitivity* of y^* to ε after the addition of a stoichiometric quantity of fuel for all three of the systems described in this paper. The original system is highly sensitive to the disturbance, as is evident from the dependence of y^* on ε. In contrast, the compensated and streamlined systems are completely insensitive to the disturbance, indicating that these systems are robust to competition from waste strands.

It appears that a biochemical nanodevice, even a very simple nanomotor, requires a control system to regulate its performance and increase its robustness. For example, the

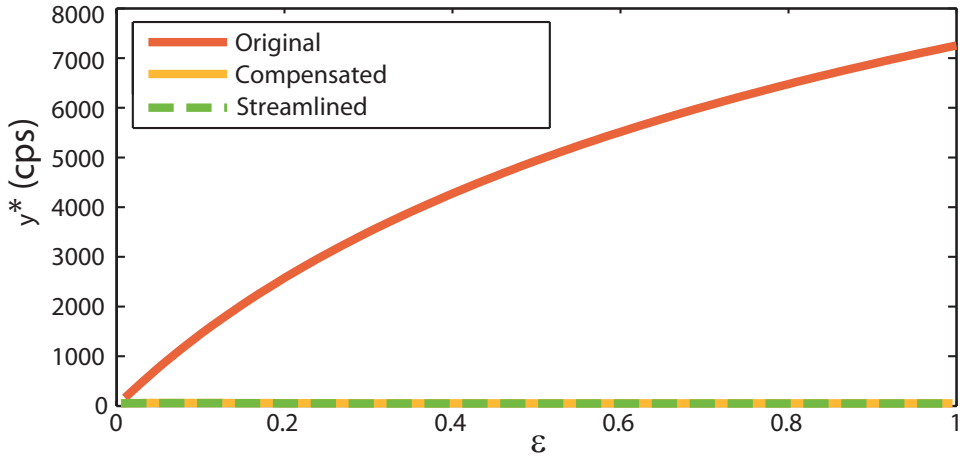

Figure 3.5: Plot of the sensitivity of nanomotor performance to the effect of competition from waste strands after a single addition of fuel. The performance metric, y^*, is a measure of the difference between the base fluorescence level and the equilibrium fluorescence. The dynamics for each nanomotor system are described by the equation $\dot{\mathbf{v}} = \mathbf{A}\cdot\hat{\mathbf{K}}(\mathbf{v})+\varepsilon\mathbf{A}\cdot\tilde{\mathbf{K}}(\mathbf{v})$, where $\mathbf{A}\cdot\hat{\mathbf{K}}(\mathbf{v})$ models the ideal behavior of a nanomotor system, and $\mathbf{A}\cdot\tilde{\mathbf{K}}(\mathbf{v})$ models the competition from waste strands. In the ideal case, where accumulating waste has no effect on nanomotor operation, $y^* = 0$. Although the performance of the original system is very sensitive to competition from waste strands, the performance of the compensated and streamlined systems is completely insensitive to this competition.

pathways resulting from the inclusion of R_H to the DNAzyme nanomotor system can be considered to be a rudimentary control system. We speculate that more complex, integrated, and interconnected nanodevices will require equally complex control systems that govern their behaviors. In fact, we next consider the problem of dynamically regulating the RNA fuel supply for this and other such devices, a consideration made possible by the introduction of transcriptional switches [49], or genelets.

3.6 Materials and Methods

3.6.1 DNA Oligonucleotides

All oligonucleotides were purchased, purified and lyophilized, from Proligo (Sigma-Aldrich). The purification method was reverse-phase, high-performance liquid chromatography (RP-HPLC). Table 3.2 contains the sequences of the DNA and RNA oligonucleotides used in experiments. Upon receipt, DNA and RNA oligonucleotides were vortexed, resuspended to 100 μM in TAE/Mg^{2+} buffer (40 mM Tris, pH 8.0, 20 mM acetic acid, 2 mM EDTA and 12.5 mM $Mg(Ac)_2$), and aliquoted into 0.6 μL reaction vials to be stored at -20°C until used.

Table 3.2: DNA and RNA sequences used in DNA nanomotor experiments.

Strand	Sequence[1]
E	5'-GGTTAGATGGTATGCTTCGGACAGGCTAGCTACAACGAGAGTGACTGAGCGTAA GGTCTGG-3'
$\bar{E}$	5'-GGTTAGATGGTATGCTTCGGACATGAATGACTGAGCGTAAGGTCTGG-3'
F	5'-Rhodamin Green-GCATACCATCTAACCTCCAGACCTTACGCTC-BHQ-1-3'
S	5'-rGrUrCrArUrUrCrArUrGrUrCrCrGrA-3'
S_1	5'-rGrUrCrArUrUrCrA-3'
S_2	5'-rUrGrUrCrCrGrA-3'

[1]The notation rN indicates a ribonucleotide base, where N is A, C, U, or G.

Note that the work done by Chen, Wang, and Mao [12] involved a chimeric RNA/DNA fuel strand and the fluorophore/quencher pair TET and TAMRA. Here, an RNA fuel strand and the fluorophore/quencher pair of Rhodamin Green and Black Hole Quencher-1 were used.

3.6.2 Nanomotor Formation

DNA strands E (or $\bar{E}$ for the streamlined system) and F were combined at an equal molar ratio in TAE/Mg^{2+} buffer to a final concentration of 0.5 μM. The nanomotor complex M was formed by annealing this solution on the following schedule: 95°C (3 min), 65°C (30 min), 50°C (30 min), 37°C (30 min) and 22°C (30 min). The solution of M at 0.5 μM was then used immediately or stored at -20°C.

3.6.3 FRET Analysis

FRET experiments were started with 600 μL of M at 0.5 μM in a 1 mL spectrofluorometry cuvette (Starna). Additionally, the compensated and streamlined systems were started with 50 units of R_H, *E. coli* (MBI-Fermentas). For all systems, a stoichiometric quantity of RNA fuel strand S was periodically added to a solution of M, followed by rapid pipette mixing for at least 7 seconds. In the streamlined system, 50 units of R_H was also added with every fifth addition of fuel. Fluorescence emission data were recorded on a Jobin-Yvon SPEX spectrofluorometer equipped with a Peltier heating/cooling device (Wavelength Electronics). All data were collected at 22°C with slit widths of 5 nm. The samples were excited at 504 nm and the emission photon count was collected at 531 nm for 1 second every 10 seconds.

Chapter 4

A ROBUST, TUNABLE, FEEDBACK-REGULATED GENELET SYSTEM – DESIGN

4.1 Introduction

Biological systems operate in highly uncertain environments, preserving functionality and performance regardless of changing external conditions. The concept of homeostasis is often invoked to describe how biological systems accomplish this feat through the robust regulation of internal conditions. Unfortunately, current synthetic biological systems are often not robust in the same fashion, although there is much interest in engineering such systems that approach the complexity of, and that may interface with, natural biological systems [3]. We believe that such systems will require the homeostatic regulation of key species in order to maintain functionality and performance as their architectural and environmental complexities increase.

Many engineered biological systems are based on rationally-designed DNA and RNA molecules [21, 89, 94, 121, 115, 122, 80, 86, 85] due to the inherent biological flexibility of nucleic acids, the increasing predictive power of nucleic acid design tools [17], and the decreasing costs of oligonucleotide synthesis [10]. A major drawback of these systems, however, is their dependence on various nucleic acid "fuel" strands or complexes, which they require to drive forward their desired dynamic behavior. For example, the depletion of these fuels over time can impact functionality and degrade performance due to changes in system dynamics. For another, the similarity of these nucleic acid fuels to other nucleic acid system components increases the likelihood of spurious interactions, which may also impact functionality and degrade performance [8].

Synthetic, *in vitro* transcriptional switches [49], or *genelets*, provide the framework for engineering regulated, *in vitro*, synthetic biological systems in a bottom-up fashion [51, 29, 52, 106]. Genelet networks display dynamic behavior driven by enzymatic RNA

polymerization and degradation reactions rather than nucleic acid fuel interactions, yet the interactions of system components can still be programmed via nucleic acid sequence selection. Another advantage of genelet networks is that they may approximate the architectural and functional complexity of naturally occurring genetic regulatory networks without the environmental complexity of cellular systems. Also, genelets operate at temperatures and in buffer conditions conducive to the operation of other engineered, *in vitro* synthetic biological systems, unlike competing *in vitro* regulatory frameworks [71]. Therefore, as genelet technology continues to develop, homeostatically regulated genelet systems may become the basis for complex synthetic systems that interact in predictable ways with natural biological systems.

Here we describe the rational design a tunable, synthetic, genelet-based, RNA concentration regulator. The regulator is a negative-feedback control device based on an autoregulation motif that has been widely studied as a common in vivo genetic regulatory motif [96] due to its simplicity, performance [87], and robustness [5, 88]. In particular, we implement this negative-feedback control device with a self-regulated genelet, which delivers RNA *fuel* for consumption by downstream devices. We also implement a similar, but unregulated, RNA fuel delivery device, which by comparison shows that negative-feedback results in a robustness to varying levels of activity in downstream molecular devices.

A detailed description of the genelet systems can be expressed in a variety of ways. Linguistically, the vocabulary of molecular biology is best suited to an informal description, while the various programming and scripting languages and notations associated with software like Visual DSD [55] and NUPACK [118, 17, 19, 20, 119] are useful for formal descriptions. Visually, *in vitro* DNA systems are often depicted with individual strands as colored lines or series of letters, with small harpoons or text direction indicating $5'$-$3'$ directionality; complexes as aligned strands; and reactions as directed arrows with reactant strands and complexes on the left-hand side, and products on the right-hand side as in chemical reaction networks (CRNs), or reactant-catalyzed reactions on the right-hand side as in genetic regulatory networks (GRNs). We will rely on several of these methods to develop and describe design specifications and models for genelet systems at a variety of levels of abstraction.

We also use a common design abstraction employed in DNA circuit design: the *domain*. A domain is a contiguous sequence of nucleic acid bases that act as a single functional subunit. Thus, a domain specifies that no two bases in the same domain will independently interact with other molecular species. We label domains with letters or numbers, and we label complementary domains the same label with and without asterisks, for example, `x` and $\texttt{x}^*$.

We first introduce our design process with a high-level, molecular-biology description of the general mechanics of genelet systems. Genelet systems consist of: nucleic acid components, including the genelets themselves; T7 RNA polymerase (R_P) and ribonuclease H (R_H) enzymes; and free ribonucleotide triphosphates, salts, and buffer conditions suitable to *in vitro* transcription reactions. A typical genelet, which is part single- and part double-stranded DNA, contains three overlapping sequence regions: a regulatory (input) region, a T7 RNAp promotor site, and a template (output) region. The regulatory region, which is entirely single-stranded, overlaps the first five bases of the promotor site. The rest of the promotor site and the template region are fully double-stranded.

A genelet is regulated either through the binding of a DNA activator strand to the regulatory domain, which completes the promoter site (activation), or through the removal and sequestration of a bound activator by an inhibitory RNA input strand (deactivation). An activated genelet generates RNA output strands from its template domain, via transcription reactions catalyzed by R_P enzymes; a deactivated genelet generates very little or no RNA output due to an incomplete promoter site.

Furthermore, genelet regulation is a dynamic process due to the highly specific conversion of DNA/RNA hybrid complexes to single DNA strands via degradation reactions catalyzed by ribonuclease H (R_H) enzymes. For example, an RNA input strand and DNA activator strand that are bound in a hybrid complex are converted by such a reaction to degraded RNA fragments and a free DNA activator strand, the latter of which is then free to interact with other RNA input strands or deactivated genelets. The ability to regulate dynamics in genelet systems makes them excellent candidates for engineering molecular control systems.

4.2 System-Level Specifications

As previously mentioned, we construct two versions of a genelet-based RNA fuel delivery device: an unregulated, or *open-loop*, device, and a negatively auto-regulated, or *closed-loop* device. The open-loop device uses a short activator, s, designed to be exactly complementary to the regulatory region of a genelet, g. This activator therefore binds g irreversibly to make a stably active genelet, $g \cdot s$, which is also a simple, unregulated source of RNA, r.

The closed-loop device uses an activator, a, which we design to be complimentary to, but longer than, the regulatory region of g. This activator therefore binds g to make g-a, which is also a source of r, and leaves an exposed *toehold* [123]. We also design r to complement and therefore sequester a into an RNA/DNA hybrid complex, $a \cdot r$. As a result, the closed-loop device $g \cdot a$ is inhibited by its own product due to a toehold-mediated, strand-displacement reaction that is initiated by r at the toehold region in $g \cdot a$. This reaction produces the sequestration complex, $a \cdot r$, and deactivated genelet g, making the closed-loop device a negatively auto-regulated source of r that can be tuned by the amount of a.

Additionally, we design r to serve as *fuel* for a downstream process. For simplicity, we design r to turn on a single-stranded DNA (ssDNA) *probe*[1][109]. The probe is a seventeen-base oligonucleotide that is fully complementary to a region of r and has a fluorophore and quencher FRET pair [114], which are incorporated at opposing ends so as to act as an output reporter in experiments. The ssDNA probe in isolation, p, coils on itself and is therefore in a closed, or *off*, conformation, as this state co-localizes the attached fluorophore and quencher pair. When p binds to r they form a rigid, double-stranded, RNA/DNA hybrid complex, $p \cdot r$, which is the open, or *on*, conformation of the probe as this state maximally separates the attached fluorophore/quencher pair. Additionally, $p \cdot r$ dynamically reverts to p via the degradation reactions catalyzed by R_H enzymes.

We call the pairing of fuel supply and downstream devices a *system*. We translate descriptions of the open- and closed-loop systems into regulatory network diagrams, which are illustrated in Figure 4.1. These diagrams are a visual representation of system-level specifications for each system, and we use them to guide the process of designing nucleic

[1] A gift from Dr. Georg Seelig.

acid sequences that implement our desired behaviors.

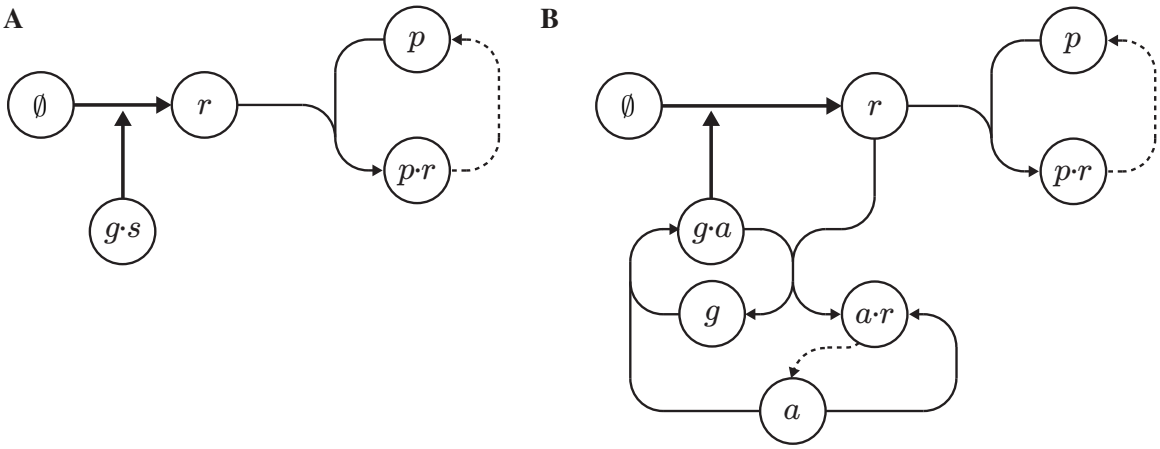

Figure 4.1: System-level design specifications, illustrated as regulatory network diagrams. (**A**) In the open-loop system, an activated DNA genelet ($g \cdot s$) catalytically produces RNA fuel, (r), via transcription reactions initiated by a T7 RNA polymerase enzyme (bold lines). RNA fuel opens a closed DNA-based molecular probe (p). An open probe ($p \cdot r$) catalytically degrades to a closed probe via a reaction initiated by an ribonuclease H enzyme (dashed lines). (**B**) In the closed-loop system, a deactivated genelet (g) is activated by a DNA activator strand with a toehold (a). An activated genelet can react with RNA fuel via a toehold-mediated strand invasion reaction, implementing negative feedback. This genelet–product reaction produces an RNA-sequestered activator complex ($a \cdot r$) and a deactivated genelet. An RNA-sequestered activator complex degrades into a free activator strand.

4.3 Domain-Level Specifications

To physically implement the systems with synthesized nucleic acids, we must choose the sequences of the nucleic acid components in each system so they determine all – and only – the interactions that are specified by our system-level specifications, which are illustrated in Figure 4.1 and described above. We begin this process by translating the system-level specifications into the domain-level specifications illustrated in Figure 4.2, one specified interaction at a time. We design both systems to use the same g to 1) simplify the design process, 2) use a single probe design, and 3) ensure the systems perform comparably in experimental tests.

We first design the sequence of the input region of g, which implements two interactions:

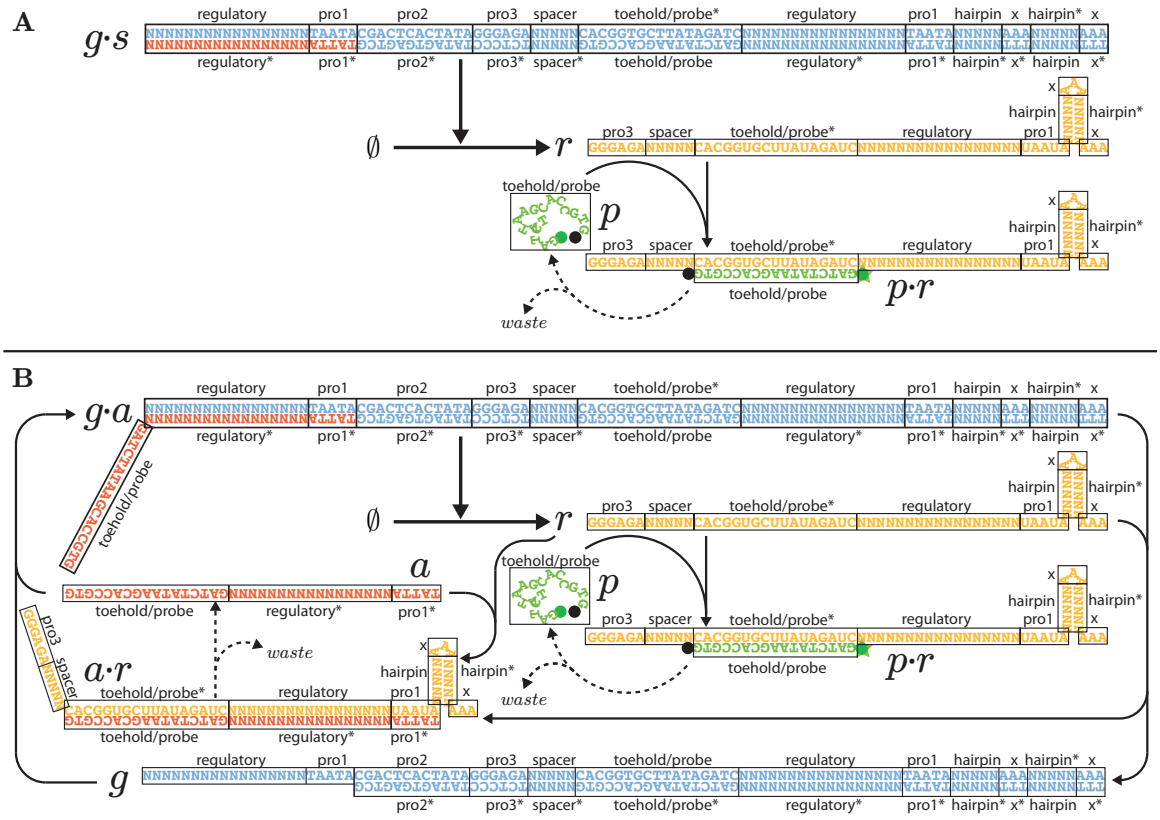

Figure 4.2: Domain-level design specifications, based on the regulatory network diagrams in Figure 4.1. Reactions: transcription (bold lines), degradation (dashed lines), and nucleic acid hybridization (thin lines). (**A**) Open-loop system. Components: activated DNA genelet ($g \cdot s$), RNA fuel (r, yellow letters), closed probe (p, green letters) labeled with a fluorophore–quencher pair (green and black dots, respectively), and open probe ($p \cdot r$) with active fluorphore (green star). (**B**) Closed-loop system. Components: activated DNA genelet ($g \cdot a$), DNA activator with toehold (a, red letters), RNA fuel (r, yellow letters), RNA-sequestered activator complex ($a \cdot r$), deactivated genelet (g, blue letters), closed probe (p, green letters) labeled with a fluorophore–quencher pair (green and black dots, respectively), and open probe (pr) with active fluorphore (green star). (Note: Left-to-right reading order of nucleic acid sequences indicates $5'$ to $3'$ directionality and the symbol $*$ represents complementarity. The symbol `N` represents an unconstrained nucleotide.)

the binding of s or a, and the completion of the $5'$-end of the promoter sequence by s or a at the overlap with the promoter region. The domains `regulatory` and `pro1`, respectively, represent the ssDNA sequences that implement these interactions. The `regulatory` sequence and length are not constrained, but do determine the specificity to s and a as well their binding rate; we set the length to 17 nucleotides to ensure fast binding. The `pro1` sequence is constrained by the promoter sequence, for which we use the canonical T7 RNA polymerase promoter sequence. The `pro1` length is only constrained in multiple genelet systems, where the degree of promoter inactivation (greater with longer `pro1`) must be balanced against the possibility of non-specific promoter activation (greater with longer `pro1`). Although this constraint does not apply to single-genelet systems, we maintain the established convention of using five nucleotides from the $5'$-end of the promoter sequence.

The core of the promoter region implements a single interaction: R_P binding. The domains `pro2` and `pro2`* represent the dsDNA sequence that implements this interaction. The length and sequence are determined by the part of the promoter sequence that does not overlap with the input or output regions. The domains `pro3` and `pro3`* represent the dsDNA sequence that overlaps the promoter and output regions, and are determined by the location of the transcription start site in the promoter sequence.

We next include a "spacer" sequence to ensure that abortive transcripts are inert. The domain `spacer` contains the sequence (along with `pro3`) that should be considered inert should transcription of r end early. We define abortive transcripts to be sequences less than 11 nucleotides in length, which constrains the `spacer` length to five nucleotides.

Most importantly, the requirements of the feedback architecture of the closed-loop genelet system, in which r interacts with both a and p, dictate that significant sequence similarity must exist between these nucleic acid components. We address the concern that p might interact with components other than r by choosing the toehold region of a to have identical sequence to p. This choice ensures that p does not interact with a, s, or g, since p has no significant secondary structure. The domain `toehold/probe`* represents this sequence, and is therefore constrained to the sequence and length of p. The implementation of the feedback loop in the closed-loop system is completed by including the `regulatory` and `pro1` domains upstream of the `toehold/probe`* domain in the output region of g.

Finally, to ensure that transcription terminates correctly, we include sequences at the $3'$-end of r that form a hairpin loop structure during transcription. The domains `hairpin` and `x` represent the sequences necessary to form a five-bp stem, three-nt loop structure with a three-nt tail. We have now defined the necessary structure and sequence constrains to design and analyze sequences using NUPACK [118, 17, 19, 20, 119], a software package with nucleic acid sequence design and thermodynamic analysis modules.

4.4 Sequence Design

We encode all of our design choices and constraints, illustrated in Figure 4.2, in the following NUPACK script:

```
# Secondary structure constraints
structure a  = ......................................
structure ar = ..........(((((((((((((((((((((((((((((((((((
               (((((((...)))))...+)))))))))))))))))))))))))))
               ))))))))))
structure g  = ....................(((((((((((((((((((((((((((
               ((((((((((((((((((((((((((((((((((((((((((((((((
               ((((((+))))))))))))))))))))))))))))))))))))))))))
               ))))))))))))))))))))))))))))))))))))))))))))))))
structure ga = ((((((((((((((((((((((((((((((((((((((((((((((((
               ((((((((((((((((((((((((((((((((((((((((((((((((
               ((((((+))))))))))))))))))))))))))))))))))))))))))
               ))))))))))))))))))))))))))))))))))))))))))))+)))))))
               ))))))))))))))).................
structure p  = ................
structure pr = ..........(((((((((((((((.................
               ....(((((...)))))...+))))))))))))))))
structure r  = ........................................
               ...(((((...)))))...
```

```
9
10 # Domin constraints
11 domain regulatory = NNNNNNNNNNNNNNNNN
12 domain pro1 = TAATA
13 domain pro2 = CGACTCACTATA
14 domain pro3 = GGGAGA
15 domain spacer = NNNNN
16 domain toeholdprobe = GATCTATAAGCACCGTG
17 domain hairpin = NNNNN
18 domain x = AAA
19
20 # Domain constraints
21 a.seq  = pro1* regulatory* toeholdprobe
22 ar.seq = pro3 spacer toeholdprobe* regulatory pro1 hairpin x
          hairpin* x pro1* regulatory toeholdprobe
23 g.seq  = regulatory pro1 pro2 pro3 spacer toeholdprobe*
          regulatory pro1 hairpin x hairpin* x x* hairpin x*
          hairpin* pro1* regulatory* toeholdprobe spacer*
          pro3* pro2*
24 ga.seq = regulatory pro1 pro2 pro3 spacer toeholdprobe*
          regulatory pro1 hairpin x hairpin* x x* hairpin x*
          hairpin* pro1* regulatory* toeholdprobe spacer*
          pro3* pro2* pro1* regulatory* toeholdprobe
25 p.seq  = toeholdprobe
26 pr.seq = pro3 spacer toeholdprobe* regulatory pro1 hairpin x
          hairpin* x toeholdprobe
27 r.seq  = pro3 spacer toeholdprobe* regulatory pro1 hairpin x
          hairpin* x
28
29 # Stop conditions
```

```
30 a.stop[%]  = 1.0
31 ar.stop[%] = 1.0
32 g.stop[%]  = 1.0
33 ga.stop[%] = 1.0
34 p.stop[%]  = 1.0
35 pr.stop[%] = 1.0
36 r.stop[%]  = 1.0
37
38 # Design parameters
39 material = dna
40 temperature[C] = 37.0
41 sodium[M] = 0.05
42 magnesium[M] = 0.012
43 trials = 10
44 prevent = AAAA, CCCC, GGGG, UUUU, KKKKKK, MMMMMM, RRRRRR,
           SSSSSS, WWWWWW, YYYYYY
```

Lines 2–8 encode the secondary structure constraints written in *dot-parens* notation. In this notation, the symbols "`.`", "`(`", and "`)`" represent single nucleotides and the symbol "`+`" represents the concatenation of separate nucleic acid strands. The keyword `structure` defines a named structure consisting of one or more component nucleic acid strands, with unpaired nucleotides denoted by the "`.`" symbol and paired nucleotides denoted by "`(`" and "`)`" symbols of equal scope. We only define the structures corresponding to the closed-loop system components a, $a{\cdot}r$, g, $g{\cdot}a$, p, $p{\cdot}r$, and r, since the open-loop sequences are a subset of the closed-loop sequences.

Lines 11–18 encode the domain sequence and length constraints. The keyword `sequence` defines a named domain with the given sequence and length constraints consisting. The structure and sequence constraints are then threaded together on lines 21–27, using the ⟨*structure*⟩**.seq** command.

Lines 30–36 define stop conditions for the design algorithm using the ⟨*structure*⟩**.stop** command, which specifies the allowed normalized ensemble defect in the sequence design for each named structure. The normalized ensemble defect calculates the average percentage of incorrectly paired bases between the target structure and the Boltzmann-weighted ensemble of structures that form at equilibrium for a particular sequence design.

Lines 39–44 pass parameters to the design algorithm. Energy parameters are not available for mixed RNA/DNA systems, so we choose to treat r as DNA in the context of the NUPACK software. We must also ignore the role of R_P and R_H in this context, except to ask for designs at a temperature and in salt conditions consistent with enzyme-dependent experimental conditions. Finally, we ask for ten sequence designs and prevent certain sequence patterns from appearing in the designs.

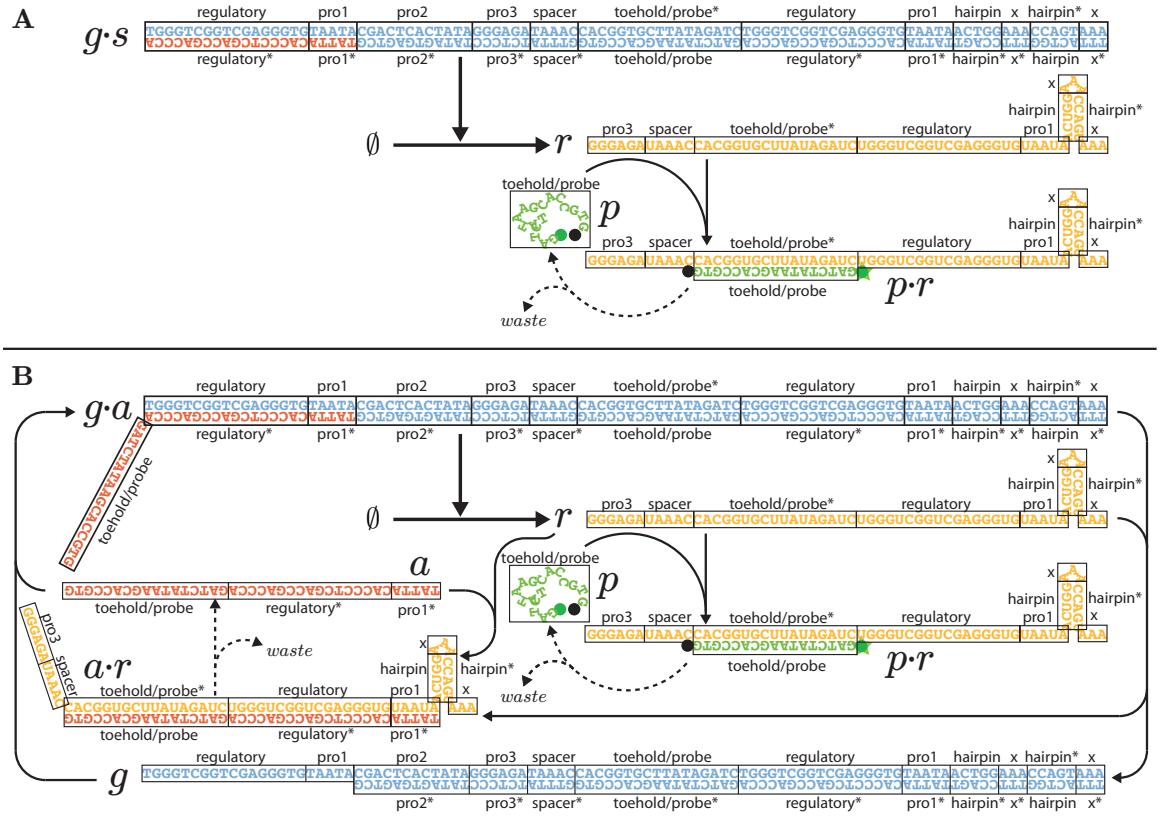

Figure 4.3: Final design specification of the open- and closed-loop genelet systems.

4.5 Sequence Analysis

The "best" sequence design from the ten generated by this script in the NUPACK design module is illustrated in Figure 4.3. To determine the "best" design, we evaluate each sequence design in the NUPACK analysis module, which provides information on the ensemble structures formed at equilibrium, for a given design, in a dilute solution. We impose two ad-hoc measures on the quality of each design by evaluating the sequences in a nominal system ($[g] = [a] = [p] = 100$ nM), under four different "initial conditions" ($[r] = 0$, 50, 100, and 200 nM), with a maximum structure size of four strands. First, we look at the maximum concentration of any incorrect structure in each ensemble. Second, we examine the graph of ensemble pair fractions, which depicts the fraction of structures in the ensemble containing a particular nucleotide pairing, charted for all nucleotides.

Table 4.1: Results of NUPACK analysis of "best" sequence design.

Initial Conditions (nM)				Equilibrium Conditions (nM)							
$[a]$	$[g]$	$[p]$	$[r]$	$[a]$	$[a{\cdot}r]$	$[g]$	$[g{\cdot}a]$	$[p]$	$[p{\cdot}r]$	$[r]$	[other]
100	100	100	0	0.03748	-	0.03748	99.96	99.99	-	-	< 0.004
100	100	100	50	-	49.99	49.99	49.99	99.99	-	-	< 0.006
100	100	100	100	-	99.99	99.97	-	99.99	-	-	< 0.008
100	100	100	200	-	99.99	99.96	-	0.3537	99.61	0.3536	< 0.012

The results of the analysis is summarized in Table 4.1 and in Figure 4.4. The last column of Table 4.1 contains the maximum concentration of any incorrect structure, which is predicted by NUPACK for each of the four analysis cases to be on order of picomolars or less. Also, the ensemble pair fractions plotted in Figure 4.4 show that very few incorrect bases form with any great probability in any of the four analysis cases. These analyses, although based on predictions made for thermodynamic equilibrium, provide a basic check that the correct structures will form over a range of dynamic equilbria, such as in our analysis cases $[r] = 0$, $[r] < [p]$, $[r] = [p]$, and $[r] > [p]$. Thus we obtain a reasonable degree of confidence that synthesized strands with the sequences in Figure 4.4 will fold as desired.

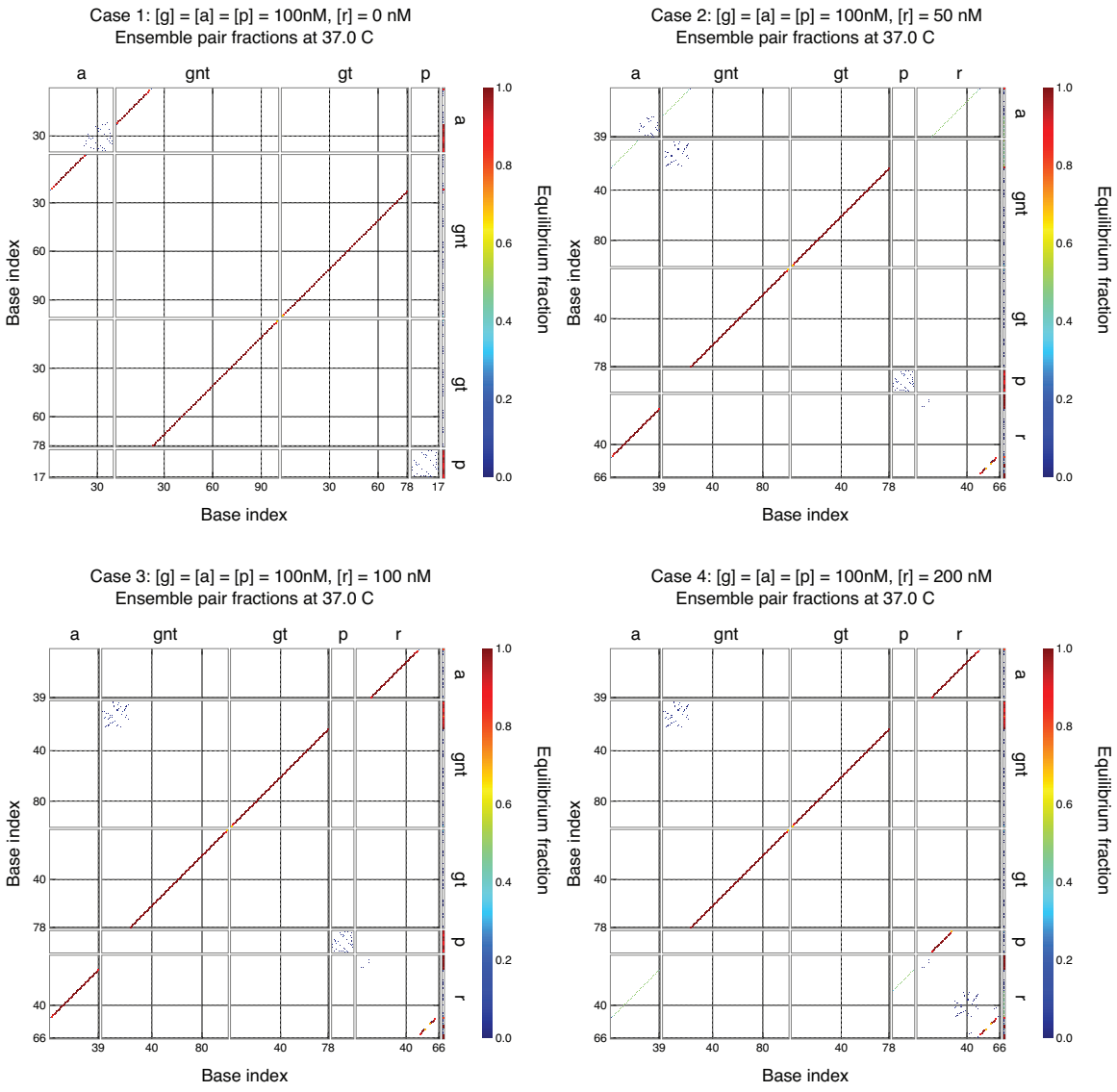

Figure 4.4: Ensemble pair fractions, calculated by NUPACK during analysis of the design in Figure 4.3, for each of the cases in Table 4.1.

Chapter 5

A ROBUST, TUNABLE, FEEDBACK-REGULATED GENELET SYSTEM – MODELS

5.1 Introduction

Our principal goal in constructing the closed-loop device is to build a regulated fuel supply that is insensitive to disturbances downstream, or at least less sensitive than a comparable unregulated fuel supply. A simple test of this property is to increase the number of downstream devices consuming fuel, then observe the effect on the supply of fuel. Thus we ask: How sensitive is the concentration of r to changes in the total amount of probe, which we denote p^{tot}, in each type of system?

We perform a mathematical analysis on simple models of the open- and closed-loop systems to estimate whether our design achieves our goal. We begin by developing a general form for models of the open- and closed-loop systems from the system-level specifications in Figure 4.1. We first consider the case in which enzyme activities are represented by Michaelis-Menten approximations, which allows us to probe the behavior of the dynamic equilibrium in open-loop with respect to system parameters. We also consider the simpler case in which enzyme activities are represented by first-order, mass-action approximations from which we derive, via a time-scale separation, approximate models for the time dependence of $[r]$ in both systems. We then perform a sensitivity analysis on these approximate models to better understand the effect of changing p^{tot} on equilibrium $[r]$, which we denote as $[r]^*$.

We interpret the system-level specifications (Figure 4.1) as a set of chemical reactions. There are three types of reactions in these specifications: 1) production (transcription of r), 2) binding (nucleic acid hybridization), and 3) degradation (hydrolysis of r from $p{\cdot}r$ and $a{\cdot}r$). We assume that the binding reactions are adequately modeled by mass-action kinetics, but allow the production and degradation rates to take on arbitrary functional forms f_p and

f_d, respectively.

The general forms of the chemical reaction models are

$$\begin{array}{rcl} \emptyset & \xrightarrow{f_p([g\cdot s],[g])} & r \\ p+r & \xrightarrow{k_p} & p\cdot r \\ p\cdot r & \xrightarrow{f_d([p\cdot r])} & p \end{array} \quad \text{and} \quad \begin{array}{rcl} \emptyset & \xrightarrow{f_p([g\cdot a],[g])} & r \\ p+r & \xrightarrow{k_p} & p\cdot r \\ p\cdot r & \xrightarrow{f_d([p\cdot r])} & p \\ g\cdot a+r & \xrightarrow{k_f} & g+a\cdot r \\ g+a & \xrightarrow{k_o} & g\cdot a \\ r+a & \xrightarrow{k_a} & a\cdot r \\ a\cdot r & \xrightarrow{f_d([a\cdot r])} & a. \end{array} \tag{5.1}$$

where we make one modification to the specifications: inactive genelet may produce r at some "leak" rate. As before, the symbol g represents a deactivated genelet, s or a an activator strand, r an RNA product, and p a molecular probe. The symbol $\cdot$ represents binding, such that $g{\cdot}s$ or $g{\cdot}a$ is an activated genelet, $a{\cdot}r$ an activator strand bound by RNA, and $p{\cdot}r$ an open probe (i.e., probe bound by RNA). Reaction rate constants are denoted by k with subscripts differentiating the reaction types.

From (5.1), we obtain general forms for the open- and closed-loop dynamics, which are

$$\begin{aligned} \dot{[r]} &= f_p\left([g\cdot s],[g]\right) - k_p\,[p]\,[r] \\ \dot{[p\cdot r]} &= k_p\,[p]\,[r] - f_d\left([p\cdot r]\right) \end{aligned} \quad \text{and} \quad \begin{aligned} \dot{[r]} &= f_p\left([g\cdot a],[g]\right) - k_p\,[p]\,[r] \\ &\quad -k_f\,[g\cdot a]\,[r] - k_a\,[a]\,[r] \\ \dot{[p\cdot r]} &= k_p\,[p]\,[r] - f_d\left([p\cdot r]\right) \\ \dot{[g\cdot a]} &= k_o\,[g]\,[a] - k_f\,[g\cdot a]\,[r] \\ \dot{[a]} &= k_d\left([a\cdot r]\right) - k_a\,[a]\,[r] \\ &\quad -k_o\,[g]\,[a], \end{aligned} \tag{5.2}$$

Since r is the only specie in our model that degrades there are several conserved quantities within the system. Therefore, we defined the constants, in the context of the open- and closed-loop systems, respectively,

$$\begin{aligned} g^{tot} &= [g\cdot s] + [g] \\ a^{tot} &= 0 \\ p^{tot} &= [p] + [p\cdot r], \end{aligned} \quad \text{and} \quad \begin{aligned} g^{tot} &= [g\cdot a] + [g] \\ a^{tot} &= [a] + [a\cdot r] + [g\cdot a] \\ p^{tot} &= [p] + [p\cdot r], \end{aligned} \tag{5.3}$$

to represent the total amount of g, a, and p, all of which, it is important to note, are experimentally meaningful parameters.

Finally, observe that at steady-state the rates of production and degradation of r must balance in the open-loop system, since

$$\left.\begin{aligned} 0 &= f_p\left([g]^*, [s]^*\right) - k_p\,[p]^*\,[r]^* \\ 0 &= k_p\,[p]^*\,[r]^* - f_d\left([p{\cdot}r]^*\right) \end{aligned}\right\} \Rightarrow\ f_p\left([g]^*, [s]^*\right) = f_d\left([p{\cdot}r]^*\right), \tag{5.4}$$

where $[x]^*$ indicates the equilibrium concentration of x. Our intuition confirms that too much production will overwhelm degradation, saturating the output at $[p{\cdot}r] = p^{tot}$ and leading to an unstable $[r]$, but we also examine this intuition mathematically, below. This property is independent of all forms of $f_p\left([g],[s]\right)$ and $f_d\left([p{\cdot}r]\right)$, and so the open-loop system appears, in general, to be very sensitive to the concentration of enzymes. We next examine this property of the open-loop model, and other properties of both models, given specific forms of the production and degradation functions.

5.2 Model 1: Enzyme Kinetics

We can use the well-known Michaelis-Menten approximation to describe the enzyme activity in the open-loop system, such that

$$\begin{aligned} f_p([g{\cdot}s],[g]) &= \frac{k_1\,[g{\cdot}s]}{K_1\left(1+\frac{[gs]}{K_1}+\frac{[g]}{K_2}\right)} + \frac{k_2\,[g]}{K_2\left(1+\frac{[gs]}{K_1}+\frac{[g]}{K_2}\right)}, \\ f_d([pr]) &= \frac{k_3\,[p{\cdot}r]}{K_3+[p{\cdot}r]}. \end{aligned} \tag{5.5}$$

The function f_p now approximates the rate of production of r, which is due to the interaction of R_P with one of two competing substrate complexes, $g{\cdot}s$ and g. (Again, we include g as a substrate to account for leaky expression from the incomplete promoter site.) These reactions have catalysis rates k_1 and k_2, and Michaelis-Menten constants K_1 and K_2, respectively. Similarly, the function f_d now approximates the rate of degradation r, which is due to the interaction of R_H with its substrate complex, $p{\cdot}r$. This reaction has catalysis rate k_3 and Michaelis-Menten constant K_3. In general, the Michaelis-Menten approximation

defines the catalysis rate as the maximum rate of product formation, v_{max}, times the concentration of enzyme; and the Michaelis-Menten constant as the concentration of substrate that yields half the maximum rate, $v_{max}/2$.

We neglect the closed-loop system for now due to the highly non-linear impact of these functional forms on (5.2), but we can rewrite (5.2) for the open-loop system, using (5.5) and the mass-conservation relationships in (5.3). For simplicity in analysis and experiments, we consider only the case where $s^{tot} \leq g^{tot}$. We then have

$$\begin{aligned} \dot{[r]} &= \frac{c_1 + c_2\, s^{tot}}{c_3 + s^{tot}} - k_p\, \left(p^{tot} - [p{\cdot}r]\right)\, [r], \\ \dot{[p{\cdot}r]} &= k_p\, \left(p^{tot} - [p{\cdot}r]\right)\, [r] - \frac{k_3\, [p{\cdot}r]}{K_3 + [p{\cdot}r]}, \end{aligned} \tag{5.6}$$

where

$$\begin{aligned} c_1 &= \frac{K_1\, k_2\, g^{tot}}{K_2 - K_1}, \\ c_2 &= \frac{k_1\, K_2 - k_2\, K_1}{K_2 - K_1}, \\ c_3 &= \frac{K_1\, K_2 + K_1\, g^{tot}}{K_2 - K_1}, \\ c_4 &= \frac{k_1\, g^{tot}}{K_1 + g^{tot}}. \end{aligned} \tag{5.7}$$

5.2.1 Equilibrium Analysis

We now examine the equilibrium behavior of (5.6). At equilibrium,

$$\begin{aligned} 0 &= \frac{c_1 + c_2\, s^{tot}}{c_3 + s^{tot}} - k_p\, \left(p^{tot} - [p{\cdot}r]^*\right)\, [r]^* \text{ and} \\ 0 &= k_p\, \left(p^{tot} - [p{\cdot}r]^*\right)\, [r]^* - \frac{k_3\, [p{\cdot}r]^*}{K_3 + [p{\cdot}r]^*}. \end{aligned} \tag{5.8}$$

We can algebraically solve (5.8) for the equilibrium values $[r]^*$ and $[p{\cdot}r]^*$, which are

$$\begin{aligned} [r]^* &= \frac{c_5\, \left(c_5 - c_6\, k_3\right)}{k_p\, c_6\, \left(c_5\, K_3 + \left(c_5 - c_6\, k_3\right)\, p^{tot}\right)} \text{ and} \\ [p{\cdot}r]^* &= \frac{c_5\, K_3}{c_6\, k_3 - c_5}, \end{aligned} \tag{5.9}$$

where

$$c_5 = c_1 + c_2\, s^{tot} \text{ and}$$
$$c_6 = c_3 + s^{tot}. \tag{5.10}$$

We can clearly see from (5.9) that the steady-state output of the system, $[p \cdot r]^*$, is independent of the term p^{tot}. Thus, the output of the system $[pr]$, when a stable steady-state exists, will demonstrate a behavior known to biologists as "perfect adaptation," alternately known as disturbance rejection, with respect to disturbances in p^{tot}. However, as we see in experiments, the observation of this phenomenon is dependent on the output mapping, where here we assume perfect observation, i.e. $y = [p \cdot r]$. We discuss this issue more in Chapter 6.

The existence of a stable steady-state requires positive values of $[r]^*$ and $[p \cdot r]^*$, however, which depends on the system parameters and initial conditions. However, for a given set of parameters, we can predict the stable steady-states of the system using the equilibrium model (5.9). To illustrate this property, we take as an example a system with specific parameters and initial conditions.

5.2.2 Results

The values in Table 5.2.2 represent realistic, but hand-picked, parameters based on previous genelet studies [49] and initial conditions based on current experimental design. This set is of interest due to the saturation of the system output, $[p \cdot r]$, at p^{tot} when increasing the initial condition s^{tot}, as illustrated in Fig. 5.1 and discussed below.

Table 5.1: Set of parameters and initial conditions for equilibrium analysis of the open-loop enzyme kinetics model.

Parameters (units)	k_1 (nM/h)	k_2 (nM/h)	k_3 (nM/h)	K_1 (nM)	K_2 (nM)	K_3 (nM)	k_p (/nM/h)	g^{tot} (nM)	p^{tot} (nM)
	150	150	145	10	100	10	0.1	100	100

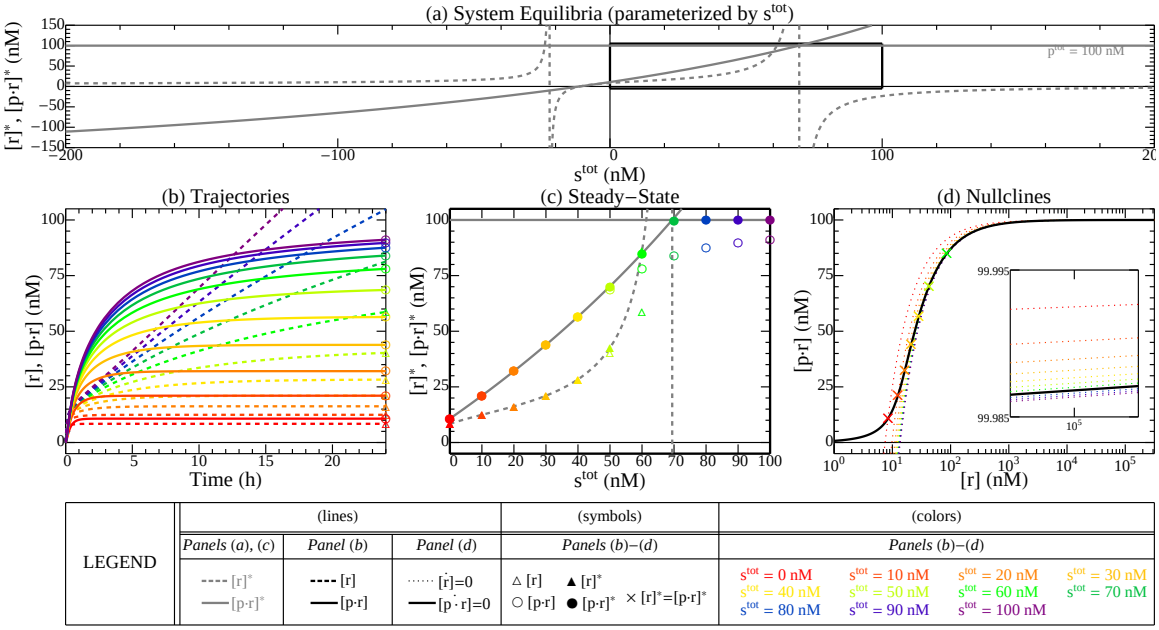

Figure 5.1: Simulations of the system model using the parameters in Table 5.2.2. (a) Plot of system equilibria parameterized by s^{tot}, which shows the region in which $[r]^* > 0$ and $[pr]^* > 0$ where the system has a stable steady-state. Here, this region corresponds to s^{tot} in the interval $[0, 69.4444)$. The bold black box corresponds to experimental conditions, where we vary s^{tot} in the interval $[0, 100]$. (b) Plot of $[r]$ and $[p{\cdot}r]$ trajectories over time, with varied s^{tot} in the interval $[0, 100]$ as specified in the legend. (c) Magnified plot of system equilibria parameterized by s^{tot}, overlaid with points $\left(s^{tot}, [r]\right)$ and $\left(s^{tot}, [p{\cdot}r]\right)$ taken at $t = 24$ h from simulated trajectories, and points $\left(s^{tot}, [r]^*\right)$ and $\left(s^{tot}, [p{\cdot}r]^*\right)$ calculated for $t \to \infty$. (d) Nullclines overlaid with intersecting points $([r]^*, [p{\cdot}r]^*)$ for each trajectory yielding a stable steady state. (d, Inset) Magnification showing the asymptotic approach of the non-stable trajectories' nullclines, which never intersect.

First, we observe in the parameterization of (5.9) by s^{tot} plotted in Fig. 5.1(a) that the output saturation of $[p{\cdot}r]^*$ at p^{tot} coincides with a point of discontinuity in $[r]^*$. We can calculate from (5.9) that when $([r]^*, [p{\cdot}r]^*) = (\infty, p^{tot})$, then

$$s^{tot} = \frac{c_1 \left(K_3 + p^{tot}\right) - c_3\, k_3\, p^{tot}}{k_3\, p^{tot} - c_2 \left(K_3 + p^{tot}\right)}, \tag{5.11}$$

which for this example is $s^{tot} = 69.4444$ nM. This bifurcation between stable and unstable behavior occurs in the physical system only when the expression (5.11) is positive, as in this

example.

Second, we plot several simulated system trajectories with varied initial condition s^{tot} in Fig. 5.1(b) to show the effect of this bifurcation more clearly. Fig 5.1(c) charts endpoint measurements of $[r]$ and $[p{\cdot}r]$ for these trajectories for two time points: 1) realistic experimental $t_f = 24$ h and 2) equilibrium $t_f \to \infty$; note that the former set of points approach the latter. The seven trajectories with initial condition s^{tot} in the interval $[0, 69.4444)$ have stable steady-states predicted by (5.9) while those four with initial condition s^{tot} in the interval $(69.4444, 100]$ have $[r]$ increasing to infinity while $[p{\cdot}r]$ saturates at p^{tot}.

Third, as a final characterization of the equilibrium behavior of this example, we examine the nullclines of the full model (5.6)

$$\begin{aligned}
[p{\cdot}r] &= p^{tot} - \frac{c_5}{k_p\,[r]\,c_6}, \\
[p{\cdot}r] &= \frac{-k_3 - k_p\,\left(K_3 - p^{tot}\right)\,[r]}{2\,k_p\,[r]} \\
&\quad \pm \frac{\sqrt{4\,K_3\,p^{tot}\,k_p^2\,[r]^2 + \left(k_3 + k_p\,\left(K_3 - p^{tot}\right)\,[r]\right)^2}}{2\,k_p\,[r]},
\end{aligned} \tag{5.12}$$

which correspond to $[\dot{r}] = 0$ and $[\dot{p{\cdot}r}] = 0$, respectively. The steady-states of the stable trajectories correspond to the points of intersection of their nullclines, plotted for this example in Fig. 5.1(d). The nullclines of the unstable trajectories never meet, and the inset of Fig. 5.1(d) shows the lack of intersection even at large $[r]$ for the four unstable trajectories, as expected.

The equilibrium analysis of the open-loop model emphasizes the role that finding useful initial conditions will play in experimental tests. The initial concentrations of R_P, R_H, g^{tot}, s^{tot} (and, presumably, a^{tot} in the closed-loop system), and p^{tot} impact the existence of a steady-state, which is necessary for testing the properties of each system that we wish to confirm. We discuss this issue more in Chapter 6.

5.3 Model 2: Mass-Action Kinetics

We now simplify the chemical models in (5.1) to use first-order rate constants to approximate enzyme activity. This approximation embodies a trade-off between simplicity and reality

in our models, which we leverage to make qualitative statements about the behavior of the systems without knowledge of parameter values. With this assumption, the chemical reaction models in (5.1) now become equivalent to

$$\begin{array}{rcl} g{\cdot}s & \xrightarrow{k_{r_1}} & g{\cdot}s + r \\ g & \xrightarrow{k_{r_2}} & g + r \\ p + r & \xrightarrow{k_p} & p{\cdot}r \\ p{\cdot}r & \xrightarrow{k_{d_1}} & p \end{array} \quad \text{and} \quad \begin{array}{rcl} g{\cdot}a & \xrightarrow{k_{r_1}} & g{\cdot}a + r \\ g & \xrightarrow{k_{r_2}} & g + r \\ p + r & \xrightarrow{k_p} & p{\cdot}r \\ p{\cdot}r & \xrightarrow{k_{d_1}} & p \\ g{\cdot}a + r & \xrightarrow{k_f} & g + a{\cdot}r \\ g + a & \xrightarrow{k_o} & g{\cdot}a \\ r + a & \xrightarrow{k_a} & a{\cdot}r \\ a{\cdot}r & \xrightarrow{k_{d_2}} & a. \end{array} \tag{5.13}$$

which we convert, by the laws of mass-action, to models of the open- and closed-loop dynamics

$$\begin{aligned} \dot{[r]} &= k_{r_1}[g{\cdot}s] + k_{r_2}[g] - k_p[p][r] \\ \dot{[p{\cdot}r]} &= k_p[p][r] - k_{d_1}[p{\cdot}r] \end{aligned} \quad \text{and} \quad \begin{aligned} \dot{[r]} &= k_{r_1}[g{\cdot}a] + k_{r_2}[g] - k_p[p][r] \\ &\quad -k_f[g{\cdot}a][r] - k_a[a][r] \\ \dot{[p{\cdot}r]} &= k_p[p][r] - k_{d_1}[p{\cdot}r] \\ \dot{[g{\cdot}a]} &= k_o[g][a] - k_f[g{\cdot}a][r] \\ \dot{[a]} &= k_{d_2}[a{\cdot}r] - k_a[a][r] \\ &\quad -k_o[g][a], \end{aligned} \tag{5.14}$$

5.3.1 Time-scale separation

Although small relative to many *in vivo* biological models, the models (5.14)are still difficult to analyze with respect to the dynamics of r. To overcome this difficulty, we employ a method of time-scale separation that separates the "slow" dynamics of r from the "fast" dynamics of the other species [48].

We first identify system parameters that, when made sufficiently small, reduce (5.14) to an approximate solution. Here we choose a^{tot} (or s^{tot}) and p^{tot}, which produce valid approximations (as shown below), but are also physically meaningful parameters that we

can set experimentally. By (5.3), this choice implies that a, $a{\cdot}r$, $g{\cdot}a$, p, and $p{\cdot}r$ are also always small. To reflect these choices in the model, we define a change of variables using the "small" parameter ε, where

$$\begin{aligned} [s] &= \varepsilon\widehat{[s]} \\ [p] &= \varepsilon\widehat{[p]} \\ [p{\cdot}r] &= \varepsilon\widehat{[p{\cdot}r]} \end{aligned} \qquad \text{and} \qquad \begin{aligned} [a] &= \varepsilon\widehat{[a]} \\ [a{\cdot}r] &= \varepsilon\widehat{[a{\cdot}r]} \\ [g{\cdot}a] &= \varepsilon\widehat{[g{\cdot}a]} \\ [p] &= \varepsilon\widehat{[p]} \\ [p{\cdot}r] &= \varepsilon\widehat{[p{\cdot}r]} \end{aligned} \tag{5.15}$$

Additionally, we assume that the rate of transcription is much greater for an activated genelet ($k_{r_1} >> k_{r_2}$), so that $k_{r_2} = \varepsilon k_{r_1}$.

We can derive the *fast model* by first changing variables, then setting the parameter ε to zero. The fast model should contain "slow" variables that do not change with time and "fast" variables that do change with time and must converge to an asymptotically stable fixed point.

When the fast model converges asymptotically, the "fast" variables can essentially be treated as constants in the slow model. We can then derive the *slow model* by first scaling time so that

$$\tau = \varepsilon\, t, \tag{5.16}$$

changing variables, then setting the parameter ε to zero. The slow model should contain the same "slow" variables that now change with (scaled) time and the same "fast" variables that now do not change with (scaled) time.

In the following, we perform this time-scale separation on both the open- and closed-loop systems in order to obtain slow models that approximate the behavior of r as a function of r alone. We then analyze these reduced models and show mathematically that the open-loop system is more sensitive to changes in p^{tot} and furthermore that the closed-loop system can be tuned to any desired steady-state level and sensitivity to changes in p^{tot}.

Open-loop system

To derive the fast model of the open-loop system, we substitute (5.15) and $k_{r_2} = \varepsilon k_{r_1}$ into (5.2) to obtain

$$\begin{aligned} \dot{[r]} &= \varepsilon\, k_{r_1}\, \widehat{[g{\cdot}s]} + \varepsilon\, k_{r_1}\, [g] - \varepsilon\, k_p\, \widehat{[p]}\, [r] \\ \varepsilon\, \dot{\widehat{[p{\cdot}r]}} &= \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_{d_1}\, \widehat{[p{\cdot}r]}, \end{aligned}$$

which, setting $\varepsilon = 0$, reduces to the fast model

$$\begin{aligned} \dot{[r]} &= 0 \\ \dot{\widehat{[p{\cdot}r]}} &= k_p\, \widehat{[p]}\, r - k_{d_1}\, \widehat{[p{\cdot}r]}. \end{aligned} \tag{5.17}$$

where we see that, as desired, $[r]$ is unchanging in time.

We show convergence of the fast model by treating $[r]$ as a constant, r_c, and substituting the conservation equality $p^{tot} = \widehat{[p]} + \widehat{[p{\cdot}r]}$ into (5.17). This yields the ordinary differential equation

$$\dot{\widehat{[p]}} = p^{tot}\, k_p\, r_c - (k_{d_1} + k_p\, r_c)\, \widehat{[p{\cdot}r]},$$

which is asymptotically stable at the point

$$\widehat{[p{\cdot}r]}^{*} = p^{tot} \frac{k_p\, r_c}{k_{d_1} + k_p\, r_c}. \tag{5.18}$$

Therefore the dynamics of the fast model converge and we can derive the slow model.

To derive the slow model of the open-loop system, we substitute (5.15) and (5.16) into (5.2) to obtain

$$\begin{aligned} \varepsilon\, (d[r]/d\tau) &= \varepsilon\, k_{r_1}\, \widehat{[g{\cdot}s]} + \varepsilon\, k_{r_1} [g] - \varepsilon\, k_p\, \widehat{[p]}\, [r] \\ \varepsilon^2\, (d[p{\cdot}r]/d\tau) &= \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_d\, \widehat{[p{\cdot}r]}, \end{aligned}$$

which, setting $\varepsilon = 0$, reduces to

$$\begin{aligned} (d[r]/d\tau) &= k_{r_1}\, \widehat{[g{\cdot}s]} + k_{r_1} [g] - k_p\, \widehat{[p]}\, [r] \\ 0 &= k_p\, \widehat{[p]}\, [r] - k_{d_1}\, \widehat{[p{\cdot}r]}. \end{aligned}$$

This slow model of the open-loop system can be further reduced through algebraic manipulation and substitution of the conservation equalities (5.3) to yield

$$\dot{[r]} \approx g^{tot}\, k_{r_1} - p^{tot}\, D([r]) \tag{5.19}$$

where

$$D([r]) = \frac{k_{d_1}\, k_p\, [r]}{k_{d_1} + k_p\, [r]},$$

which is an approximation of the full system in terms of $[r]$ only as desired. The quality of this approximation depends on the assumption that there is a small total amount of probe in the system.

Closed-loop system

We now pursue the comparative derivation of the closed-loop system approximation. First, to derive the fast model of the closed-loop system, we substitute (5.3) and (5.15) into (5.2) to obtain

$$\begin{aligned}
\dot{[r]} &= \varepsilon\, k_{r_1}\, \widehat{[g{\cdot}a]} + \varepsilon\, k_{r_1} [g] - \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_f\, \widehat{[g{\cdot}a]}\, [r] - \varepsilon\, k_a\, \widehat{[a]}\, [r] \\
\varepsilon\, \dot{\widehat{[p{\cdot}r]}} &= \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_{d_1}\, \widehat{[p{\cdot}r]} \\
\varepsilon\, \dot{\widehat{[g{\cdot}a]}} &= \varepsilon\, k_o\, [g]\, \widehat{[a]} - \varepsilon\, k_f\, \widehat{[g{\cdot}a]}\, [r] \\
\varepsilon\, \dot{\widehat{[a]}} &= \varepsilon\, k_{d_2}\, \widehat{[a{\cdot}r]} - \varepsilon\, k_a\, \widehat{[a]}\, [r] - \varepsilon\, k_o\, [g]\, \widehat{[a]},
\end{aligned}$$

which, setting $\varepsilon = 0$, reduces to the fast model

$$\begin{aligned}
\dot{[r]} &= 0 \\
\dot{\widehat{[p{\cdot}r]}} &= k_p\, \widehat{[p]}\, [r] - k_{d_1}\, \widehat{[p{\cdot}r]} \\
\dot{\widehat{[g{\cdot}a]}} &= k_o\, [g]\, \widehat{[a]} - k_f\, \widehat{[g{\cdot}a]}\, [r] \\
\dot{\widehat{[a]}} &= k_{d_2}\, \widehat{[a{\cdot}r]} - k_a\, \widehat{[a]}\, [r] - k_o\, [g]\, \widehat{[a]},
\end{aligned} \tag{5.20}$$

where we see that, as desired, $[r]$ is unchanging in time.

We show convergence of the fast model as before, by treating $[r]$ as a constant (r_c) and substituting the conservation equalities $p^{tot} = \widehat{[p]} + \widehat{[p{\cdot}r]}$, $a^{tot} = \widehat{[a]} + \widehat{[a{\cdot}r]} + \widehat{[g{\cdot}a]}$, and $g^{tot} = [g] + \widehat{[g{\cdot}a]} \approx [g]$ into (5.20). This yields the set of ordinary differential equations

$$\begin{aligned}
\dot{\widehat{[p]}} &= (k_{d_1} + k_p\, r_c)\, \widehat{[p]} - p^{tot}\, k_{d_1} \\
\dot{\widehat{[g{\cdot}a]}} &= g^{tot} k_o\, \widehat{[a]} - k_f\, \widehat{[g{\cdot}a]}\, r_c \\
\dot{\widehat{[a]}} &= a^{tot}\, k_{d_2} - (g^{tot}\, k_o + k_{d_2} + k_a\, r_c)\, \widehat{[a]} - k_{d_2}\, \widehat{[g{\cdot}a]},
\end{aligned}$$

which has a positive, asymptotically stable steady-state at

$$\begin{bmatrix} \widehat{[p\cdot r]}^* \\ \widehat{[g\cdot a]}^* \\ \widehat{[a]}^* \end{bmatrix} = \begin{bmatrix} p^{tot}\frac{k_p\, r_c}{k_{d_1}+k_p\, r_c} \\ a^{tot}\frac{g^{tot}\, k_{d_2}\, k_o}{g^{tot}\, k_o\, k_{d_2}+\left(g^{tot}\, k_o+k_{d_2}\right) k_f\, r_c+k_a\, k_f\, r_c^2} \\ a^{tot}\frac{k_{d_2}\, k_f\, r_c}{g^{tot}\, k_o\, k_{d_2}+\left(g^{tot}\, k_o+k_{d_2}\right) k_f\, r_c+k_a\, k_f\, r_c^2} \end{bmatrix} \tag{5.21}$$

Therefore the dynamics of the fast model converge and we can derive the slow model.

To derive the slow model of the closed-loop system, we substitute (5.15) and (5.16) into (5.2) to obtain

$$\begin{aligned} \varepsilon(d[r]/d\tau) &= \varepsilon\, k_{r_1}\, \widehat{[g\cdot a]} + \varepsilon\, k_{r_1}\, [g] - \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_f\, \widehat{[g\cdot a]}, [r] - \varepsilon\, k_a\, \widehat{[a]}\, [r] \\ \varepsilon^2(d[p\cdot r]/d\tau) &= \varepsilon\, k_p\, \widehat{[p]}\, [r] - \varepsilon\, k_{d_1}\, \widehat{[p\cdot r]} \\ \varepsilon^2(d[g\cdot a]/d\tau) &= \varepsilon\, k_o\, [g]\, \widehat{[a]} - \varepsilon\, k_f\, \widehat{[g\cdot a]}\, [r] \\ \varepsilon^2(d[a]/d\tau) &= \varepsilon\, k_{d_2}\, \widehat{[a\cdot r]} - \varepsilon\, k_a\, \widehat{[a]}\, [r] - \varepsilon\, k_o\, [g]\, \widehat{[a]}, \end{aligned}$$

which, dividing through by and then setting ε to 0, reduces to

$$\begin{aligned} (d[r]/d\tau) &= g^{tot}\, k_{r_1} - k_p\, \widehat{[p]}\, [r] - k_f\, \widehat{[g\cdot a]}\, [r] - k_a\, \widehat{[a]}\, [r] \\ 0 &= k_p\, \widehat{[p]}\, [r] - k_{d_1}\, \widehat{[p\cdot r]} \\ 0 &= k_o\, [g]\, \widehat{[a]} - k_f\, \widehat{[g\cdot a]}\, [r] \\ 0 &= k_{d_2}\, \widehat{[a\cdot r]} - k_a\, \widehat{[a]}\, [r] - k_o\, [g]\, \widehat{[a]}. \end{aligned} \tag{5.22}$$

This slow model of the closed-loop system can be further reduced through algebraic manipulation and substitution of the conservation equalities (5.3) to yield

$$\dot{[r]} \approx g^{tot}\, k_{r_1} - p^{tot}\, D([r]) - a^{tot}\, K([r]) \tag{5.23}$$

where

$$K([r]) = \frac{k_{d_2}\, k_f\, (g^{tot}\, k_o\, [r] + k_a\, [r]^2)}{g^{tot}\, k_o\, k_{d_2} + (g^{tot}\, k_o + k_{d_2})\; k_f\, [r] + k_a\, k_f\, [r]^2}$$

which is an approximation of the full system in terms of r only as desired. The quality of this approximation depends on the assumption that there is a small total amount of activator and probe in the system, relative to the total amount of genelet.

5.3.2 Sensitivity analysis

We now use the approximations for the time evolution of r in (5.19) and (5.23) to compare the sensitivity of the steady-state availability of r to changes downstream. We use sensitivity

to mean the change in a given performance measure, x, when a parameter of the system, p, changes, as in

$$\begin{aligned} S_p^x &= \frac{\%\text{ change in performance}}{\%\text{ change in parameter}} \\ &= \frac{\Delta x/x}{\Delta p/p} = \frac{\Delta x}{\Delta p}\frac{p}{x} \approx \frac{\partial x}{\partial p}\frac{p}{x}. \end{aligned} \tag{5.24}$$

Since our goal is to regulate the amount of steady-state RNA available to a downstream molecular device, we define our performance measure as $[r]^*$, and our parameter of interest as p^{tot}.

Open-loop system

To find the sensitivity of $[r]^*$ with respect to p^{tot} in the open-loop system, we linearize (5.19) around the nominal value of RNA product, $[r]_0$, and nominal value of the total amount of probe, p_0^{tot}, to obtain

$$\begin{aligned} \dot{[r]} &\approx g^{tot}\,k_{r_1} - p_0^{tot}\,D(r_0) \\ &\quad - p_0^{tot}\,\frac{\partial D([r])}{\partial [r]}\bigg|_{[r]=[r]_0}([r]-[r]_0) \\ &\quad - \frac{\partial}{\partial p^{tot}}\,p^{tot}D([r]_0)\bigg|_{p^{tot}=p_0^{tot}}(p^{tot}-p_0^{tot}) \\ &\approx g^{tot}\,k_{r_1} - p_0^{tot}\,d([r]_0)([r]-[r]_0) - p^{tot}\,D([r]_0) \end{aligned}$$

where

$$d([r]_0) = \frac{\partial D([r])}{\partial [r]}\bigg|_{[r]=[r]_0}$$

Solving for equilibrium gives

$$[r]^* = \frac{g^{tot}\,k_{r_1} - p^{tot}\,D([r]_0)}{p_0^{tot}\,d([r]_0)} + [r]_0 \tag{5.25}$$

and applying the definition in (5.24) to (5.27) yields the open-loop sensitivity

$$S_{p^{tot}}^{[r]^*} \approx -\frac{D([r]_0)}{p_0^{tot}\,d([r]_0)}\frac{p_0^{tot}}{[r]_0} = -\frac{k_{d_1}+k_p\,[r]_0}{k_{d_1}}. \tag{5.26}$$

It is of principle importance to note that the open-loop sensitivity is *independent* of any experimentally-tunable parameters such as g^{tot} or a^{tot}. This result indicates that the

open-loop system is not robust to changes in p^{tot} with respect to $[r]^*$, and in fact is more sensitive the more r is delivered; and therefore makes a poor regulator of RNA fuel, as expected.

Closed-loop system

Comparatively, to find the sensitivity of r^* with respect to p^{tot} in the closed-loop system, we linearize (5.23) around r_0 and p_0^{tot} as before to obtain

$$\begin{aligned}
\dot{[r]} &\approx g^{tot}\, k_{r_1} - p_o^{tot}\, D([r]_0) - a^{tot}\, K(r_0) \\
&\quad - p_0^{tot}\, \frac{\partial D([r])}{\partial [r]}\bigg|_{[r]=[r]_0} ([r] - [r]_0) \\
&\quad - a^{tot}\, \frac{\partial K([r])}{\partial [r]}\bigg|_{[r]=[r]_0} ([r] - [r]_0) \\
&\quad - \frac{\partial}{\partial p^{tot}}\, p^{tot}\, D([r]_0)\bigg|_{p^{tot}=p_0^{tot}} (p^{tot} - p_0^{tot}) \\
&\approx g^{tot}\, k_{r_1} - p^{tot}\, D([r]_0) - a^{tot}\, K([r]_0) \\
&\quad -(p_0^{tot}\, d([r]_0) + a^{tot}\, k([r]_0))([r] - [r]_0)
\end{aligned}$$

where

$$k([r]_0) = \frac{\partial K([r])}{\partial [r]}\bigg|_{[r]=[r]_0}$$

Solving for equilibrium r gives

$$[r]^* = \frac{g^{tot}\, k_{r_1} - p^{tot}\, D([r]_0) - a^{tot}\, K([r]_0)}{p_0^{tot}\, d([r]_0) + a^{tot}\, k([r]_0)} + [r]_0 \tag{5.27}$$

and applying the definition in (5.24) to (5.27) yields the closed-loop sensitivity

$$S_{p^{tot}}^{[r]^*} \approx -\frac{D([r]_0)}{p_0^{tot}\, d([r]_0) + a^{tot}\, k([r]_0)} \frac{p_0^{tot}}{[r]_0}. \tag{5.28}$$

Here, it is of principle importance to note that the closed-loop sensitivity is *dependent* on the experimentally tunable parameter a^{tot} and that the magnitude of the sensitivity will decrease as a^{tot} increases. This equation is strong evidence that the closed-loop system matches our design specification for a robustly-regulated RNA fuel source.

5.3.3 Results

As previously stated, our principal goal in using negative feedback in the closed-loop device is to provide a more robust supply of r with respect to fluctuations in p^{tot}, the total amount of probe in the system. The time-scale separation and sensitivity analysis of the simple mass-actions models allow us to make several statements about the qualitative behavior of both systems .

First, the open-loop sensitivity of $[r]^*$ with respect to p^{tot} depends on just two rate parameters and the nominal level of $[r]$, which we denote $[r]_0$. Specifically, the magnitude of the open-loop sensitivity *increases* proportionately with $[r]_0$. Therefore, our open-loop model indicates that decreasing the effect of disturbances in p^{tot} on $[r]^*$ in a given system would require dynamic adjustment of $[r]$, a task at odds with the concept of "open-loop."

Second, although our analysis confirms the intuitive notion that dynamic self-adjustment of $[r]$ to disturbances in p^{tot} is impossible in the open-loop system, it also predicts that the open-loop sensitivity will increase as the total amount of genelet, g^{tot}, is increased. Note from Equation 5.26 that the open-loop sensitivity is proportional to $[r]_0$, which is itself proportional to g^{tot}. Therefore, open-loop systems with higher g^{tot} should be more sensitive than open-loop systems with lower g^{tot}.

Third, the magnitude of the closed-loop sensitivity of $[r]^*$ with respect to p^{tot} *decreases* proportionately with the total amount of activator, a^{tot}. Therefore, according to our closed-loop model, increasing a^{tot} in a closed-loop system would decrease the effect of disturbances in p^{tot} on $[r]^*$. So while the open-loop sensitivity cannot be tuned down as g^{tot} is increased, the closed-loop sensitivity can always be tuned down by increasing a^{tot}, regardless of the value of g^{tot}. This result supports the intuitive connection between the general ability of negative feedback loops to reject disturbances and the role of a in the feedback pathway.

Fourth, and finally, g^{tot} and a^{tot} (or s^{tot}) are all physically meaningful parameters, and can be varied across experiments. Thus, these analytical results form the basis of a testable hypothesis: closed-loop systems should produce a more robust fuel supply, with respect to changes in probe activity, compared to open-loop systems or closed-loop systems with less activator.

Chapter 6

A ROBUST, TUNABLE, FEEDBACK-REGULATED GENELET SYSTEM – TESTS

6.1 Introduction

We experimentally test the main hypothesis: closed-loop systems should produce a more robust fuel supply, with respect to changes in probe activity, compared to open-loop systems or closed-loop systems with less activator. Experimental tests of the systems monitor the total fluorescence output, typically the response of pre-mixed solutions of g and s or a, in the required buffers, to additions of enzymes and p. For our experimental testbed, we prepare and monitor experiments in the individual wells of a 96-well microplate, in a fluorescence microplate reader (see Appendix C for protocols). This testbed allows for the concurrent testing of multiple systems with varying conditions.

The mapping between the fluorescence output and concentrations of system components is one of several experimental characterizations we perform to follow up on the insights offered by our model analysis in Chapter 5. We also characterize the effect of enzyme levels and buffer conditions on system activity. The design of the nucleic acid components is contingent upon thermodynamic predictions of secondary structure and do not take into account the activity of R_P or R_H. In addition, our models indicate the systems are sensitive to initial conditions and other parameters. We look for a nominal set of conditions in which both systems respond quickly, are tunable, and do not saturate the output.

First, we examine the output mapping between the systems and the experimental testbed. Then we check the impact of initial enzyme concentrations on the equilibrium dynamics of the systems. Once we establish a feasible set of nominal initial conditions, we test our hypothesis on multiple open- and closed-loop systems. Finally, we try to connect the qualitative predictions of our mass-action model to quantitative fits of our test data.

6.2 Output Calibration

The precise output mapping from the fluorescence readings taken by the plate reader to $[p \cdot r]$ is important to our understanding of the experimental system. We present data in Figure 6.1 from calibration tests performed on p and $p \cdot r$, on two different plate readers, along with linear fits for each data set. Note that each data point is averaged over four replicates, and that the error bars are too small to see. The protocol for the data calibration is in Appendix C.

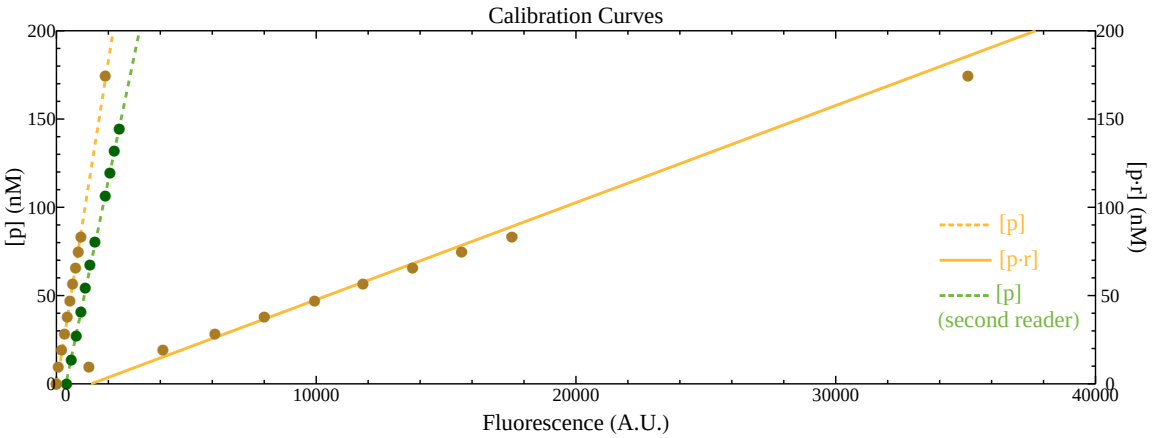

Figure 6.1: Calibration data for p and $p \cdot r$. Probe p was titrated into replicate wells containing either buffer, or buffer with r. Averaged data points were fitted to linear models for fluorescence vs. $[p]$ (dashed lines) and $[p \cdot r]$ (solid line). Error bars are smaller than the plotted points.

It is clear from the two data sets corresponding to increasing $[p]$ that the output mapping is unique to each plate reader, as p contributes a different amount of "background" fluorescence in each plate reader. Also, we note that the data set corresponding to increasing $[p \cdot r]$ falls below the associated linear fit with increased concentrations of p and r. A linear fit therefore may not properly map fluorescence values to $[p \cdot r]$, even considering background fluorescence.

The creation of an empirical mapping is made even more difficult by the fact that p is sensitive to other system components besides r. The data sets and associated linear fits plotted in Figure 6.2 correspond to increasing $[p \cdot r]$ without and with various combinations

of the system components g, s, and a. The presence of each combination of components results in a different linear fit, each of them divergent from the mapping established by measurement of $[p \cdot r]$ in isolation. The inset to Figure 6.2 shows how the fits for each combination diverge as $[p \cdot r]$ increase.

Taken together, these calibration data convince us to present data in the raw fluorescence mode, to avoid the possibility of miscounting $p \cdot r$. However, to unify models and data, we require some method for mapping fluorescence to $[p \cdot r]$. We therefore propose an output mapping, y, that defines the relationship between fluorescence, and $[p]$ and $[p \cdot r]$ as

$$y = k_{y_1} [p] + k_{y_2} p^{tot}, \tag{6.1}$$

which we use with our models when fitting data.

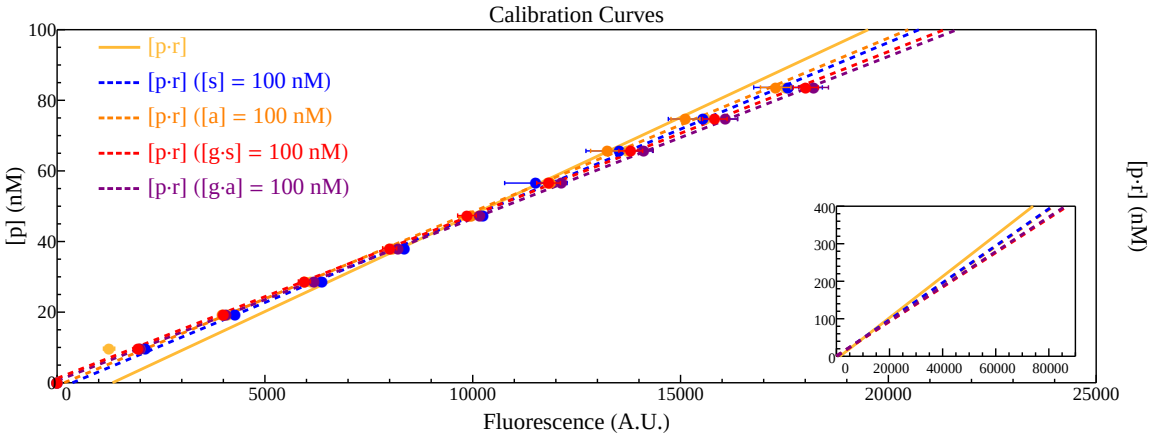

Figure 6.2: Calibration data for $p \cdot r$ in the presence of various combinations of system components g, s, and a. Probe p was titrated into replicate wells containing buffer with r and the combination of g, s, a or nothing as noted in the upper left corner of the plot. Averaged data points were fitted to linear models for fluorescence vs. $[p \cdot r]$ (solid yellow line) and $[p \cdot r]$ in the presence of system components (dashed lines). Error bars are included, except where smaller than the plotted points.

6.3 Equilibrium Dynamics

We require nominal conditions for running the open-and closed-loop systems in order to obtain a dynamic equilibrium that does not saturate p^{tot}. In the saturating regime, $[r]$ is

unstable, the closed-loop device cannot regulate r, and there is no discernible difference between the open- and closed-loop systems in the output response.

Extensive optimization of initial conditions leads us to use the nominal conditions $[g] = [s] = [a] = [p] = 100$ nM for the majority of our tests, with enzyme concentrations $[R_P] = 100$ U and $[R_H] = 1$ U. We include Figure 6.3 as an example the kind of optimization we can do in these systems. The trajectories in Figure 6.3 illustrate the effect of tuning the enzyme concentrations in the system (in this case both enzyme concentrations, while keeping a constant ratio between them). Each of the eight trajectories in Figure 6.3 depict the output response of a nominal open-loop system to which some concentration of enzymes are added after one hour.

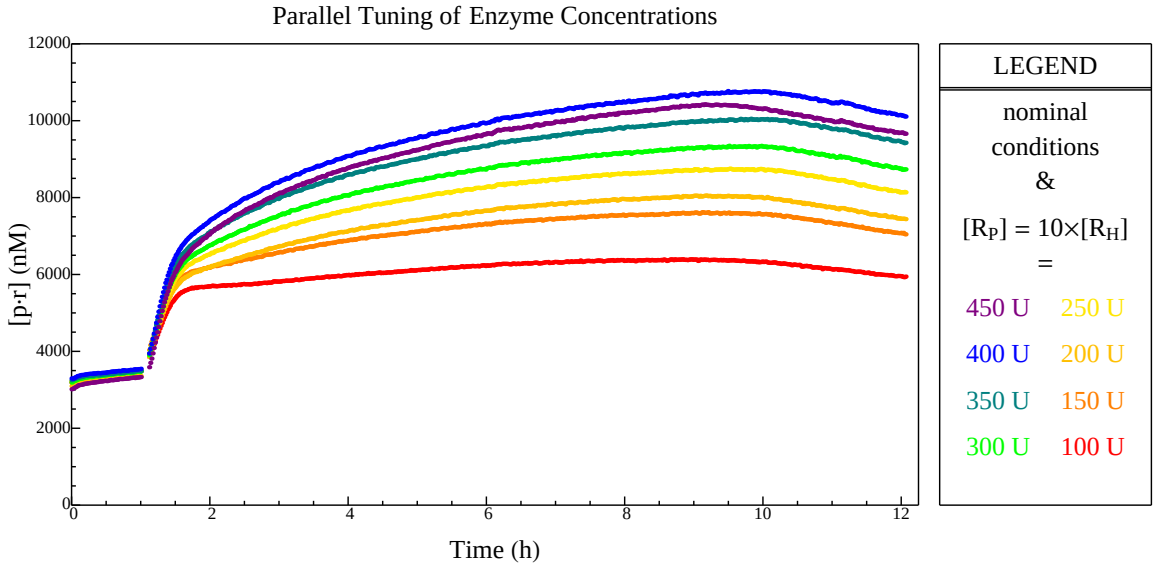

Figure 6.3: Effects of tuning enzyme concentrations with a fixed ratio in a nominal open-loop system. Enzymes at concentrations noted in the legend were added to eight open-loop systems with $[g] = [s] = [p] = 100$nM.

Note that higher enzyme concentrations lead to a slope in the equilibrium phase (lasting from one hour after enzymes are added to about nine hours after enzymes are added). The relative flatness of "slower" systems is a factor in our choice of nominal initial conditions. It is also likely that higher enzyme concentrations result in higher turnover of r, which yields

waste products from the degradation of r faster, which causes p to be more open on average. Although lower enzyme concentrations reduce the effect, faster systems–like high-feedback closed-loop systems–experience it more.

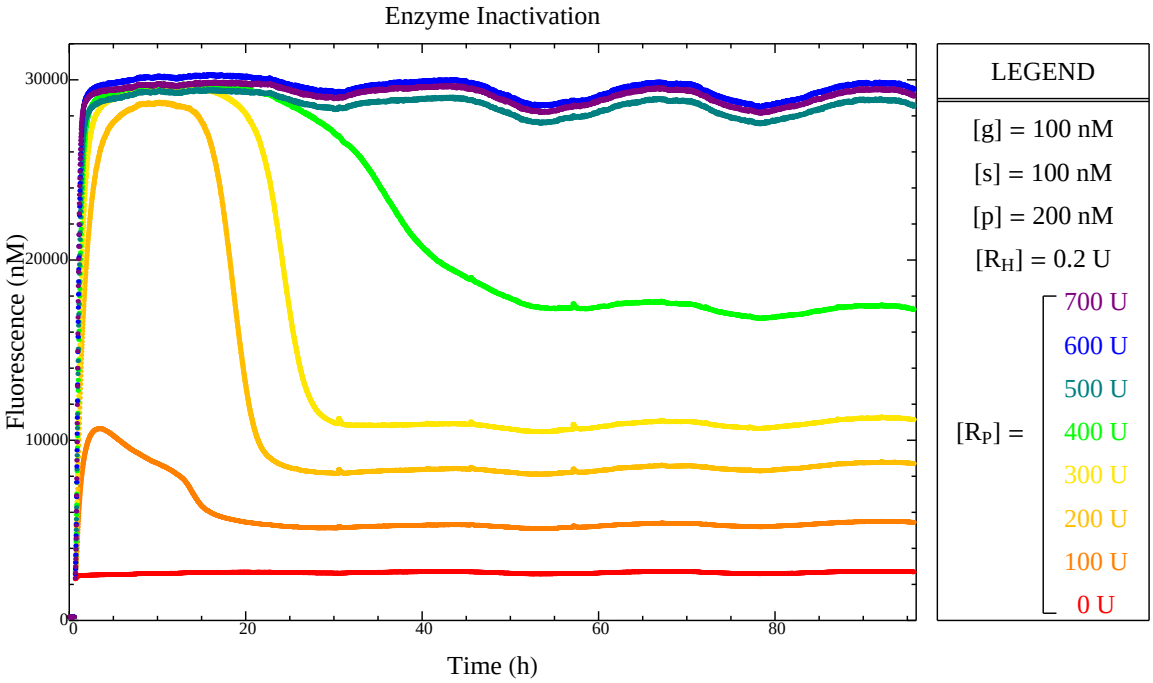

Figure 6.4: Experimental effects of enzyme inactivation in a nominal open-loop system. $[R_P]$ at concentrations noted in the legend were added to eight open-loop systems at $t = 0$h. In sub-saturating trajectories, fluorescence drops with R_P inactivation and flattens with subsequent R_H. More R_P takes longer to inactivate. In saturating regimes, fluorescence stays high due to unstable $[r]$.

Another issue with the enzymes to R_P and R_H is their inactivation over time. Figure 6.4 plots eight trajectories of an open-loop system with fixed $[R_H]$ and varying $[R_P]$. Each system response is monitored for ~90 hours after the addition of enzymes . In each case below saturation, we see the inactivation of R_P, which is marked by a drop in fluorescence, followed by the inactivation of R_H, which is marked by the lack of change in fluorescence. Since $[r]$ is unstable in saturating conditions, we cannot observe inactivation in the trajectories corresponding to $[R_P] = 500$, 600, or 700 U. One may note the small oscillations in the trajectories; these have a period of ~24 hours, and may be due to periodic temperature

changes in the room containing the plate reader.

Due to the inactivation of enzymes within ~10 hours, we test the sensitivity of the open- and closed-loop systems by changing p^{tot} at least every 2 hours. The short time period is still long enough to allow systems to reach a dynamic equilibrium, as seen in Figure 6.3, while also short enough to allow for multiple disturbances to p^{tot} in a single trajectory.

6.4 Hypothesis Validation

We test the validity of our hypothesis by directly comparing several closed-loop systems, with varying production and feedback strengths, to several open-loop systems, with varying production strengths. We expect to see that both open- and closed-loop outputs increase with g^{tot}, and we expect that closed-loop systems are more robust than open-loop systems or than other closed-loop systems with less activator.

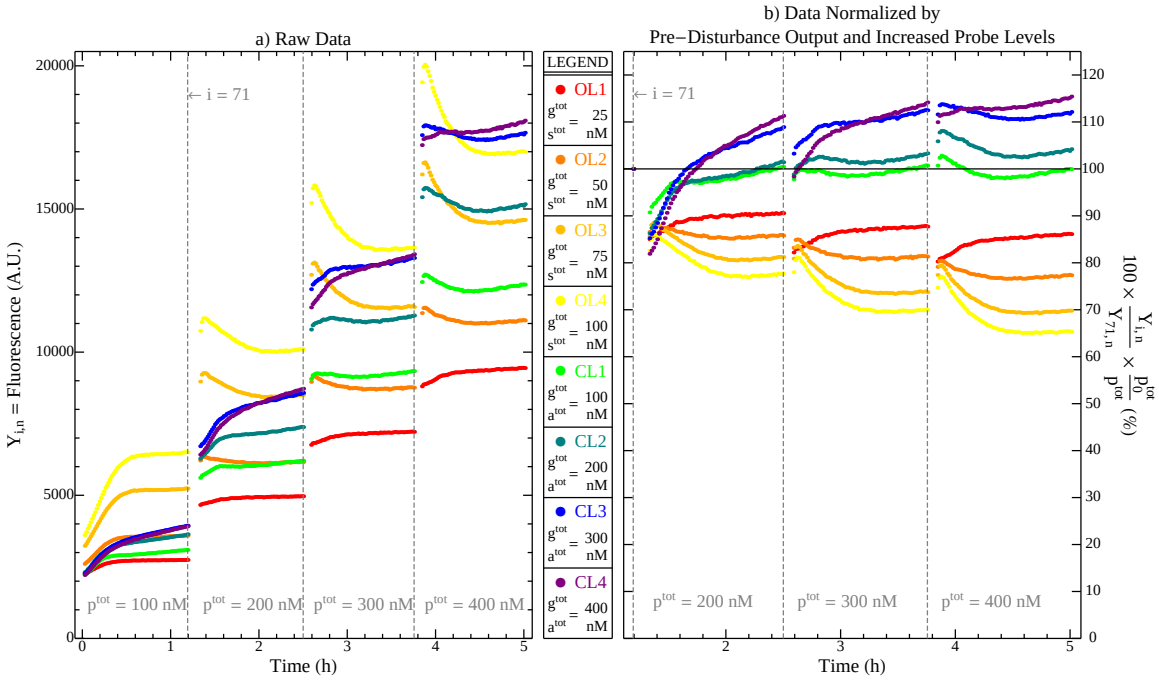

Figure 6.5:

In each experiment, disturbances in the downstream activity of p were periodically introduced through the exogenous increase of p^{tot} above the nominal level, p_0^{tot}. Multiple parallel

Table 6.1: Reagent distributions in named data sets. (Reagents were combined in batches to allow for equal distribution into each well containing a test system, and pipetting waste.)

Reagents		OL1	OL2	OL3	OL4	CL1	CL2	CL3	CL4
H_2O	vol. (μL)	66.70	66.35	66.00	65.65	65.65	64.25	62.85	61.45
-	conc. (none)	-	-	-	-	-	-	-	-
TB	vol. (μL)	10	10	10	10	10	10	10	10
10	conc. ($\times$)	1	1	1	1	1	1	1	1
rNTPs	vol. (μL)	10	10	10	10	10	10	10	10
80	conc. (mM)	8	8	8	8	8	8	8	8
$MgCl_2$	vol. (μL)	6	6	6	6	6	6	6	6
100	conc. (mM)	6	6	6	6	6	6	6	6
g (in TB)	vol. (μL)	0.1	0.2	0.3	0.4	0.4	0.8	1.2	1.6
25×10^3	conc. (nM)	25	50	75	100	100	200	300	400
s	vol. (μL)	0.25	0.5	0.75	1	-	-	-	-
10×10^3	conc. (nM)	25	50	75	100	-	-	-	-
a	vol. (μL)	-	-	-	-	1	2	3	4
10×10^3	conc. (nM)	-	-	-	-	100	200	300	400
p	vol. (μL)	1	1	1	1	1	1	1	1
10×10^3	conc. (nM)	100	100	100	100	100	100	100	100
TIPP	vol. (μL)	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
2	conc. (U/μL)	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015
RNase H	vol. (μL)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10	conc. (U/μL)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
RNAP	vol. (μL)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
1000	conc. (U/μL)	1	1	1	1	1	1	1	1
BSA	vol. (μL)	5	5	5	5	5	5	5	5
10	conc. (mg/μL)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

experiments, in which the initial conditions g^{tot} and s^{tot} or a^{tot} were varied, allowed us to compare the behavior of open- and closed-loop systems with different fuel production and negative feedback strengths.

The results of our primary hypothesis validation experiment are plotted in Figures 6.5. Each trajectory in the plot is a trajectory we denote $Y_{i,n}$, which contains fluorescence data collected in the i^{th} minute for test system n, where n is one of four open-loop systems named OL1, OL2, OL3, or OL4; or four closed-loop systems named CL1, CL2, CL3, or CL4. The initial conditions g^{tot} and s^{tot} or a^{tot} of each test system are listed in the legend of Figure 6.5. Identical disturbances to p^{tot} were introduced to each test system every 71 minutes as quickly as possible ($<$ 4 minutes). Longer periods between disturbances were impractical due to enzyme inactivation; see above.

Since quantitative time-course data is difficult to obtain for $[r]$, we looked for a way to clearly compare the sensitivities of $[r]$ to changes in p^{tot} in our test system using the data in $Y_{i,n}$. We first reasoned that a more robust supply of fuel will operate a greater percentage of downstream devices after an increase in downstream demand. We then normalized $Y_{i,n}$ by the last pre-disturbance output, $Y_{71,n}$ to directly compare systems with different $[r]_0$; and by the ratio of p^{tot} to p_0^{tot} to measure the robustness of the fuel supply. These normalized data thus show the relative robustness of the test systems, and are plotted in Figure 6.5B.

More specifically, the vertical ordering of the trajectories in Figure 6.5B corresponds to the relative robustness of the fuel supply among the test systems. The open-loop test systems all open a smaller percentage of probes in each period than the closed-loop test systems, and are therefore less robust. In addition, closed-loop test systems become more robust with higher a^{tot}, while open-loop test systems become less robust with higher g^{tot}, as expected.

It is important to note that the performance of CL2, CL3, and CL4 increases in each period. This could indicate that these systems have not reached equilibrium prior to each new disturbance. However, other experiments we performed (see, for example, the results shown in Figure 6.3) suggest that the fuel supply does stabilize within the time period, other, un-modeled processes might be responsible for a slow increase in fluorescence. For example, the ssDNA nature of p makes it particularly susceptible to nonspecific binding, such as with

the waste products of r degradation, which are in greater quantities in closed-loop systems due to the additional degradation by a.

So although the qualitative agreement of the vertical ordering of trajectories in Figure 6.5 with the predictions of the sensitivity analysis supports our basic hypothesis that, in contrast to the open-loop device, the closed-loop device is a robust, tunable source of r, we next assessed the quantitative agreement of our model to our data $Y_{i,n}$. This assessment required optimizing the set of free parameters, θ in the model such that simulations of the model with optimized parameters, θ^*, fit the data.

Table 6.2: The set of model parameters to be fit for each data set, an initial guess at the best fit parameter set, and the range of values that each parameter can take.

Free Parameter	Initial Guess	Range
k_{r_1} (/h)	10	$(10^{-3}, 10^{4})$
k_{r_2} (/h)	10^{-1}	$(10^{-5}, 10^{2})$
k_{d_1} (/h)	10	$(10^{-6}, 10^{1})$
k_{d_2} (/h)	10	$(10^{-6}, 10^{1})$
k_a (/nM/h)	10^{-1}	$(10^{-6}, 10^{2})$
k_f (/nM/h)	10^{-1}	$(10^{-6}, 10^{2})$
k_o (/nM/h)	10^{-1}	$(10^{-6}, 10^{2})$
k_p (/nM/h)	10^{-1}	$(10^{-6}, 10^{2})$
k_{y_1} (A.U./nM/h)	10^{2}	$(10^{-3}, 10^{3})$

6.5 Data Fitting

We employ a data-fitting approach [36] that was also successfully applied to a positive feedback genelet system [106]. We use the software package `SloppyCell` [36] to generate ensembles of model parameters sets consistent with our test data. The software takes as inputs an SBML version of our model (Appendix B), our data, and the initial model parameter set shown in Table 6.2. It first optimizes the cost separately for each test system,

starting from the initial parameter set, then generates an ensemble of model parameter sets, for each test system, that is consistent with the data.

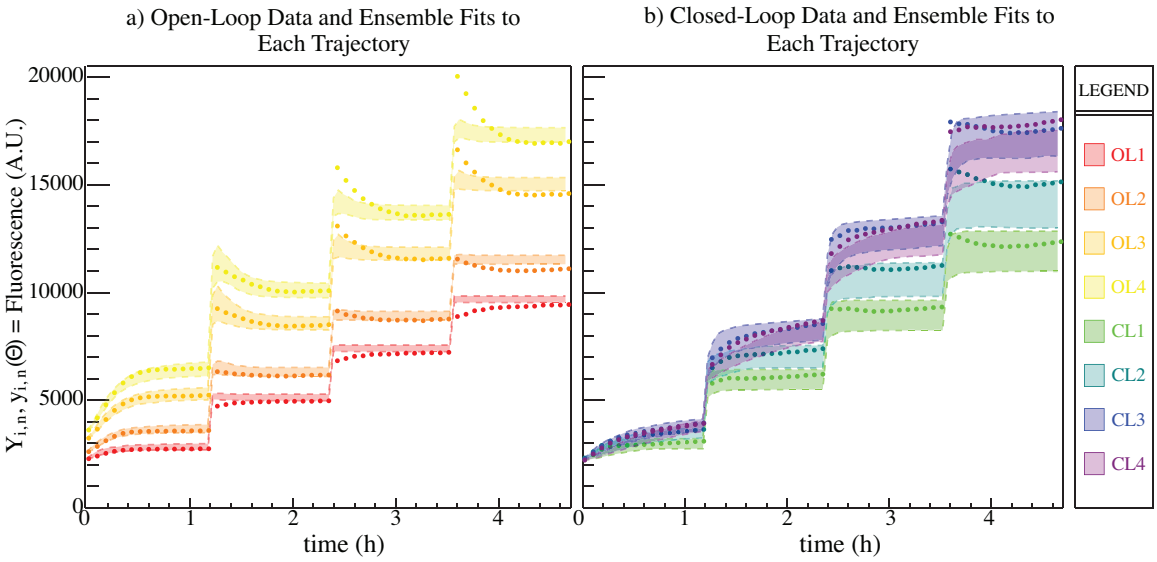

Figure 6.6: Individual data trajectories (dots) and individual `SloppyCell` model fits (95% confidence intervals; translucent windows).

Briefly, each ensemble contains a distribution of parameter sets, θ, consistent with the probability density

$$P(\theta) \propto \exp^{C(\theta,T)/T}$$

where $T = 1$ is the temperature and C is a cost function of the form

$$C(\theta) = \frac{1}{2}\sum_i \left(\frac{B_{n,i}\, y_{n,i}(\theta) - Y_{n,i}}{\sigma_{n,i}}\right) + \text{priors}$$

for system n, model prediction $y_{n_i}(\theta)$, "scale factor" $B_{n,i}$, data point $Y_{n,i}$ and uncertainty $\sigma_{n,i}$ of that data point. The 'priors' term

$$\frac{1}{2}\left(\frac{\log\theta_i - \log\theta_i^*}{\sigma_{\log\theta_i}}\right)^2$$

keeps individual parameters θ_i from extending more than $\sigma_{\log\theta_i}$ from θ_i^*.

Figures 6.6A and 6.6B plot the open- and closed-loop test data, respectively, along with the 95% confidence interval consistent with the ensembles, which were individually

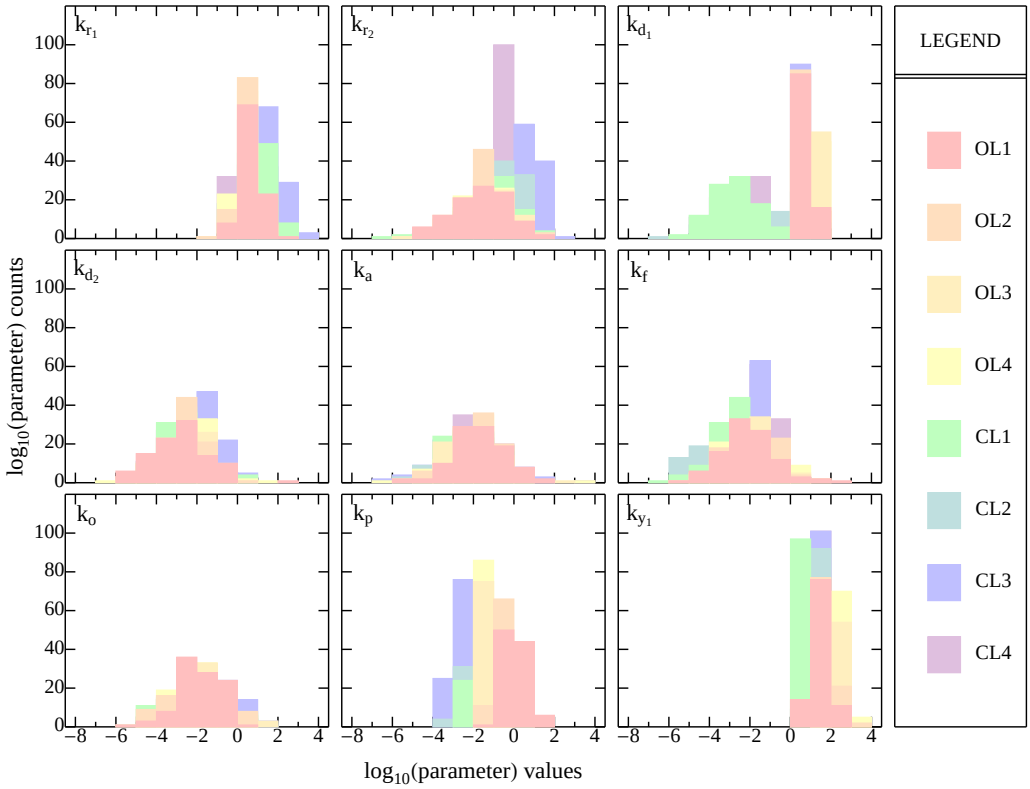

Figure 6.7: Parameter distributions of individual `SloppyCell` model fits, color-coded by system.

generated for each test system. Each interval represents a collective fit specific to one data set. While these model fits generally contain most of the data trajectories, they do not predict all of the quantitative features in the data. Particularly, the transient features of each collective fit are less exaggerated than are the transient features of the data trajectory in later periods.

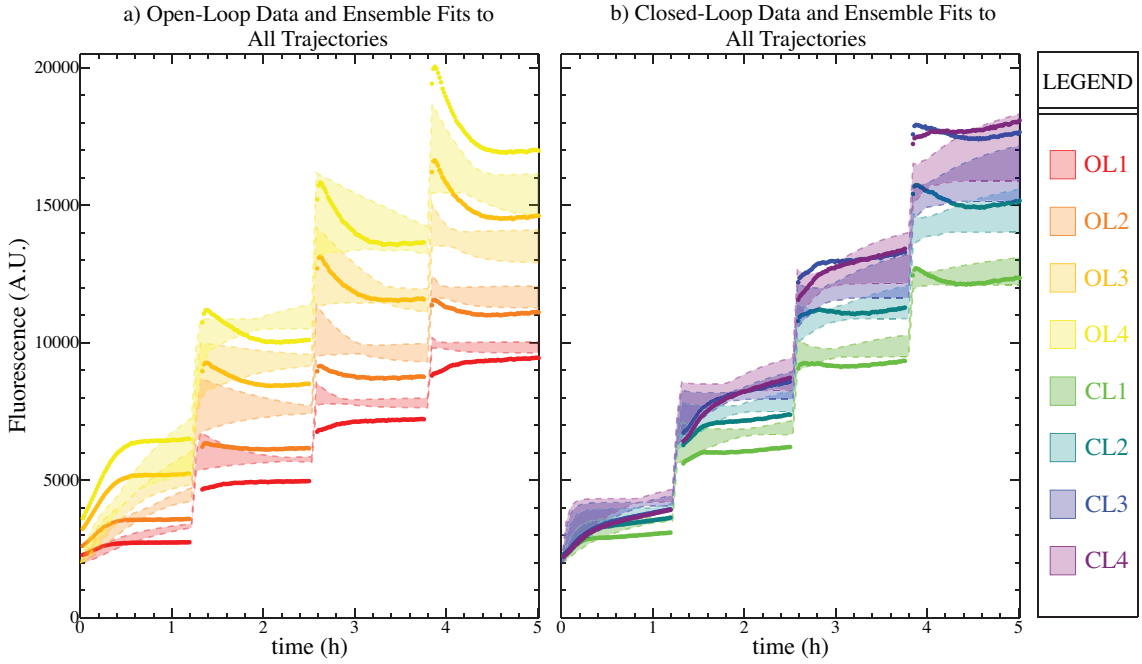

Figure 6.8: Data trajectories (dots) and `SloppyCell` model fit (95% confidence intervals; translucent windows) to all trajectories at once.

Figure 6.7 plots the distributions of individual parameter values within each ensemble. The distributions of model parameters k_{r_1}, k_{r_2}, k_{d_1}, and k_p vary the most among the ensembles. We might expect this for k_{r_1}, k_{r_2}, and k_{d_1}, since we have used them in our simple model to approximate higher-order enzyme processes, but the variation in the distribution of k_p might indicate "sloppiness" in the model [36].

We also plot the 95% confidence interval on the model fit to all the data at once. This collective fit, as one might expect, performs less well than individually fitting each data set. Both the steady-state and the transient dynamics are generally unmatched by the fit for

any data trajectory in any time period.

Since the ensembles that yield these collective fits are generated near θ^*, the quality of the fits is limited by the cost function landscape local to θ^*. It seems clear from Figures 6.6, 6.7, and 6.8 that there are multiple minima in the cost function landscape, and local methods suffer from a poor choice of θ^*.

In Figures 6.9 and 6.10, we plot the results of an alternative fitting method, in which we generate ensembles consistent with

$$P(\theta) \propto \exp^{C(\theta,T)/T}$$

but randomly, rather than directed by local search methods. Figure 6.9 shows the data trajectories for OL1, OL2, OL3, and OL4 plotted with the hundred best model fits, for each data set, found by this random Bayesian ensemble algorithm. Figure 6.10 shows the same for CL1, CL2, CL3, and CL4. In each case, the model fits collectively form a window that contains the actual data trajectory.

Despite the quality of these fits, the fact remains that the model is poorly constrained by the data. Figure 6.11 plots the distributions of each parameter in each random Bayesian ensemble. The relatively flat distribution of all parameters across all systems illustrates the fact that cost function landscape either flat, or contains many local minima.

6.6 Discussion

Robust, tunable, dynamic regulation of RNA fuel is achievable with even the simplest feedback architecture, as we have demonstrated. As the complexity of systems that rely on DNA-based nanodevices grow, it will be increasingly evident that tight regulation of fuel and other intermediate components is necessary for the robust operation of such systems. Indeed, natural biological systems point the way, as researchers uncover more and more elegant regulatory schemes from the cellular level up to the physiological. Synthetic biologists have already demonstrated, in vitro, DNA-based computation [94, 123, 86], assembly [21, 115], signal processing [80, 104], and regulatory [49, 29, 28, 106] systems of scalable complexity. Combining these systems into, for example, a synthetic cell will require appropriate control architectures to ensure robust behavior.

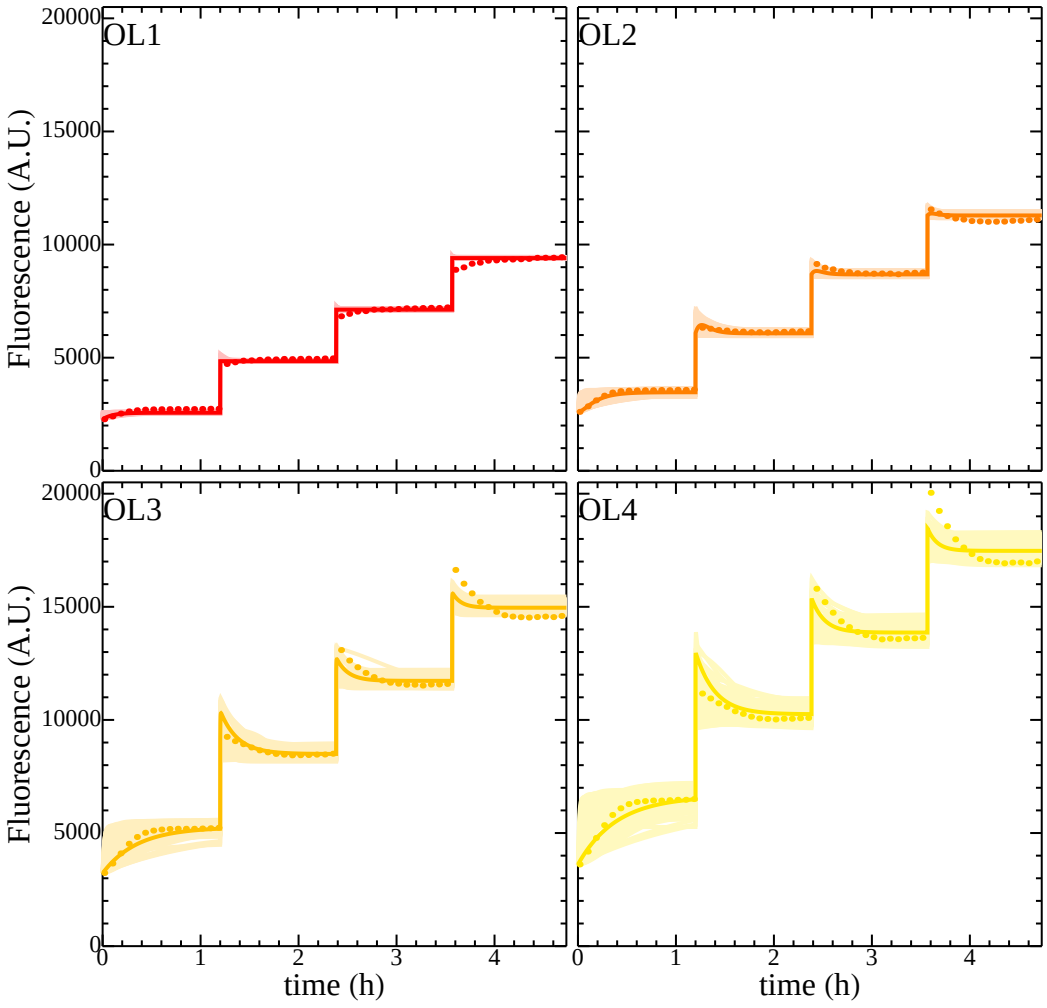

Figure 6.9: Open-loop data trajectories (dots), random Bayesian ensemble model fits (light-colored lines), and best random Bayesian ensemble model fit (dark solid line).

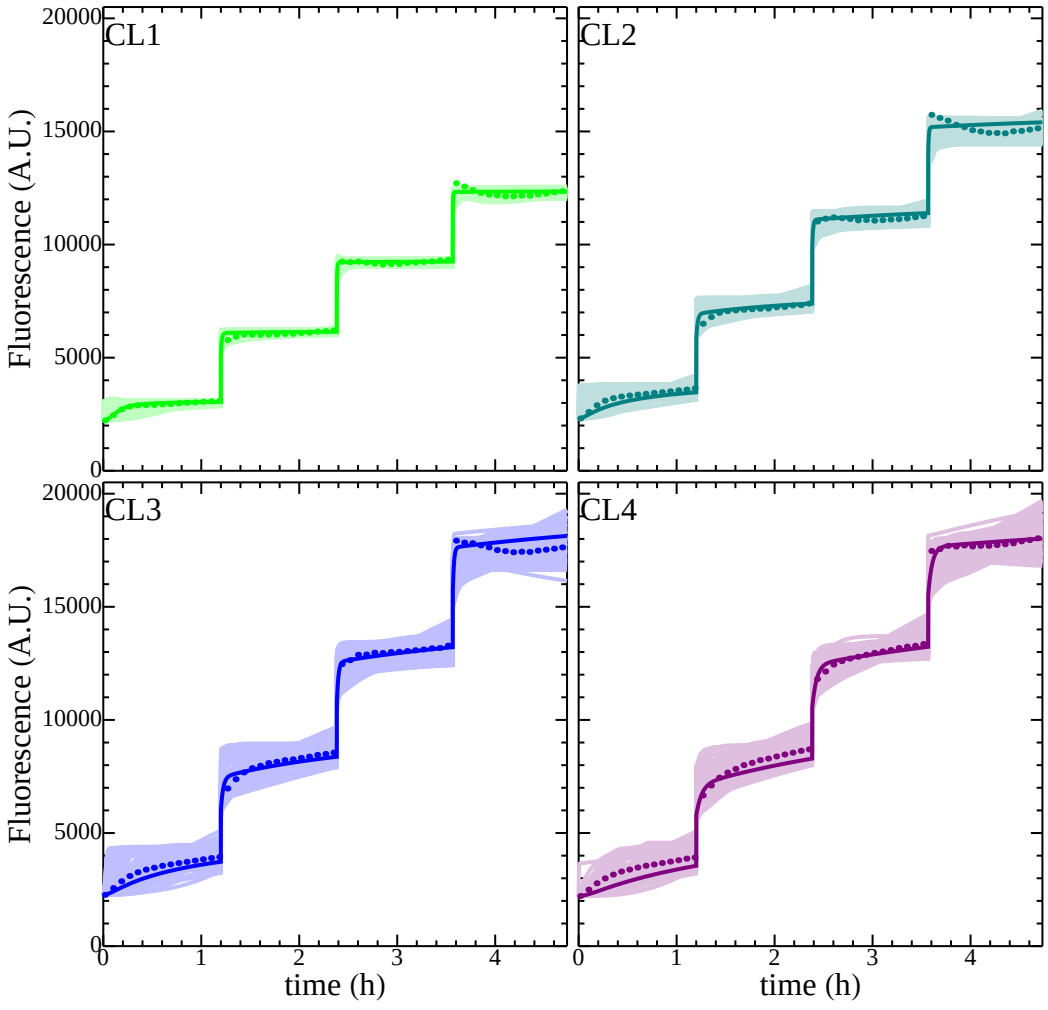

Figure 6.10: Closed-loop data trajectories (dots), random Bayesian ensemble model fits (light-colored lines), and best random Bayesian ensemble model fit (dark solid line).

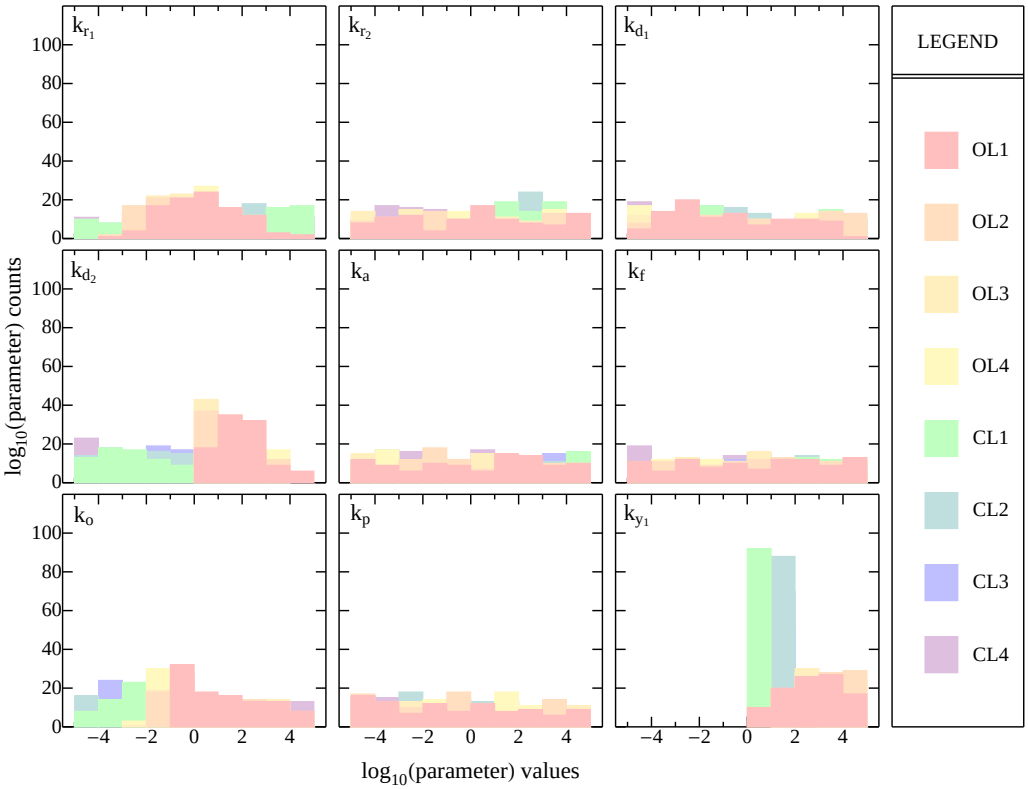

Figure 6.11: Parameter distributions of individual random Bayesian model fits, color-coded by system.

BIBLIOGRAPHY

[1] Registry of standard biological parts. `http://partsregistry.org`.

[2] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, 2:2006.0028, Jan. 2006.

[3] D. Baker, G. Church, J. Collins, D. Endy, J. Jacobson, J. Keasling, P. Modrich, C. Smolke, and R. Weiss. Engineering Life: Building a FAB for Biology. *Scientific American*, 294(6):44–51, June 2006.

[4] J. Bath and A. J. Turberfield. DNA nanomachines. *Nature Nanotechnology*, 2(5):275–84, May 2007.

[5] A. Becskei and L. Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590–3, June 2000.

[6] S. A. Benner and A. M. Sismour. Synthetic biology. *Nature Reviews. Genetics*, 6(7):533–43, July 2005.

[7] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for Your Mathematical Plays*. Academic Press, London, 1982.

[8] J. D. Bishop and E. Klavins. An improved autonomous DNA nanomotor. *Nano Letters*, 7(9):2574–7, Sept. 2007.

[9] I. Budin and N. K. Devaraj. Membrane assembly driven by a biomimetic coupling reaction. *Journal of the American Chemical Society*, 134(2):751–3, Jan. 2012.

[10] R. Carlson. The pace and proliferation of biological technologies. *Biosecurity and Bioterrorism : Biodefense Strategy, Practice, and Science*, 1(3):203–14, Jan. 2003.

[11] R. H. Carlson. *Biology Is Technology*. Harvard University Press, 2010.

[12] Y. Chen, M. Wang, and C. Mao. An Autonomous DNA Nanomotor Powered by a DNA Enzyme. *Angewandte Chemie International Edition*, 43(27):3554–3557, 2004.

[13] R. Chhabra, J. Sharma, Y. Liu, and H. Yan. Addressable molecular tweezers for DNA-templated coupling reactions. *Nano Letters*, 6(5):978–83, May 2006.

[14] H. M. T. Choi, J. Y. Chang, L. A. Trinh, J. E. Padilla, S. E. Fraser, and N. A. Pierce. Programmable in situ amplification for multiplexed imaging of mRNA expression. *Nature Biotechnology*, 28(11):1208–12, Nov. 2010.

[15] David Yu Zhang. Towards Domain-Based Sequence Design for DNA Strand Displacement Reactions. In Y. Sakakibara and Y. Mi, editors, *DNA Computing and Molecular Programming*, volume 6518 of *Lecture Notes in Computer Science*, pages 162–175, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[16] B. Ding and N. C. Seeman. Operation of a DNA robot arm inserted into a 2D DNA crystalline substrate. *Science*, 314(5805):1583–5, Dec. 2006.

[17] R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic Analysis of Interacting Nucleic Acid Strands. *SIAM Review*, 49(1):65, Jan. 2007.

[18] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–403, Jan. 2004.

[19] R. M. Dirks and N. A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, 24(13):1664–77, Oct. 2003.

[20] R. M. Dirks and N. A. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry*, 25(10):1295–304, July 2004.

[21] R. M. Dirks and N. A. Pierce. Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(43):15275–15278, 2004.

[22] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. MacMillan, July 1992.

[23] A. J. Dzieciol and S. Mann. Designs for life: protocell models in the laboratory. *Chemical Society Reviews*, 41(1):79–85, Jan. 2012.

[24] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, Jan. 2000.

[25] D. Endy. Foundations for engineering biology. *Nature*, 438(7067):449–53, Nov. 2005.

[26] A. C. Forster and G. M. Church. Towards synthesis of a minimal cell. *Molecular Systems Biology*, 2:45, Jan. 2006.

[27] A. C. Forster and G. M. Church. Synthetic biology projects in vitro. *Genome Research*, 17(1):1–6, Jan. 2007.

[28] E. Franco, E. Friedrichs, J. Kim, R. Jungmann, R. Murray, E. Winfree, and F. C. Simmel. Timing molecular motion and production with a synthetic transcriptional clock. *Proceedings of the National Academy of Sciences of the United States of America*, 108(40):E784–93, Oct. 2011.

[29] E. Franco and R. M. Murray. Design and performance of in vitro transcription rate regulatory circuits. In *2008 47th IEEE Conference on Decision and Control*, pages 161–166. IEEE, 2008.

[30] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–42, Jan. 2000.

[31] D. G. Gibson, G. A. Benders, C. Andrews-Pfannkoch, E. A. Denisova, H. Baden-Tillson, J. Zaveri, T. B. Stockwell, A. Brownley, D. W. Thomas, M. A. Algire, C. Merryman, L. Young, V. N. Noskov, J. I. Glass, J. C. Venter, C. A. Hutchison, and H. O. Smith. Complete chemical synthesis, assembly, and cloning of a Mycoplasma genitalium genome. *Science*, 319(5867):1215–20, Feb. 2008.

[32] D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie, C. Merryman, S. Vashee, R. Krishnakumar, N. Assad-Garcia, C. Andrews-Pfannkoch, E. A. Denisova, L. Young, Z.-Q. Qi, T. H. Segall-Shapiro, C. H. Calvey, P. P. Parmar, C. A. Hutchison, H. O. Smith, and J. C. Venter. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*, 329(5987):52–6, July 2010.

[33] R. Gil, F. J. Silva, J. Peretó, and A. Moya. Determination of the core of a minimal bacterial gene set. *Microbiology and Molecular Biology Reviews : MMBR*, 68(3):518–37, Sept. 2004.

[34] J. I. Glass, N. Assad-Garcia, N. Alperovich, S. Yooseph, M. R. Lewis, M. Maruf, C. A. Hutchison, H. O. Smith, and J. C. Venter. Essential genes of a minimal bacterium. *Proceedings of the National Academy of Sciences of the United States of America*, 103(2):425–30, Jan. 2006.

[35] H. Gu, J. Chao, S.-J. Xiao, and N. C. Seeman. A proximity-based programmable DNA nanoscale assembly line. *Nature*, 465(7295):202–5, May 2010.

[36] R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10):1871–78, Oct. 2007.

[37] A. Gutmann, J. W. Wagner, Y. Ali, A. L. Allen, J. D. Arras, B. F. Atkinson, N. A. Farahany, A. G. Garza, C. Grady, S. L. Hauser, R. S. Kucherlapati, N. L. Michael, and D. P. Sulmasy. New Directions: The Ethics of Synthetic Biology and Emerging Technologies. Technical report, Presidential Commission for the Study of Bioethical Issues, 2010.

[38] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl):C47–52, Dec. 1999.

[39] J. Hasty, D. McMillen, and J. J. Collins. Engineered gene circuits. *Nature*, 420(6912):224–30, Nov. 2002.

[40] P. Hausen and H. Stein. Ribonuclease H. *European Journal of Biochemistry*, 14(2):278–283, 1970.

[41] M. Heinemann and S. Panke. Synthetic biology–putting engineering into biology. *Bioinformatics*, 22(22):2790–9, Nov. 2006.

[42] F. J. Isaacs, D. J. Dwyer, and J. J. Collins. RNA synthetic biology. *Nature Biotechnology*, 24(5):545–54, May 2006.

[43] F. Jacob and J. Monod. On the Regulation of Gene Activity. *Cold Spring Harbor Symposia on Quantitative Biology*, 26(0):193–211, Jan. 1961.

[44] M. Kaern, W. J. Blake, and J. J. Collins. The engineering of gene regulatory networks. *Annual Review of Biomedical Engineering*, 5:179–206, Jan. 2003.

[45] H. Kang, H. Liu, J. A. Phillips, Z. Cao, Y. Kim, Y. Chen, Z. Yang, J. Li, and W. Tan. Single-DNA molecule nanomotor regulated by photons. *Nano Letters*, 9(7):2690–6, July 2009.

[46] E. R. Kay, D. A. Leigh, and F. Zerbetto. Synthetic molecular motors and mechanical machines. *Angewandte Chemie International Edition*, 46(1-2):72–191, Jan. 2007.

[47] A. S. Khalil and J. J. Collins. Synthetic biology: applications come of age. *Nature Reviews. Genetics*, 11(5):367–79, May 2010.

[48] H. K. Khalil. *Nonlinear Systems (3rd Edition).* Prentice Hall, 2001.

[49] J. Kim. Ph.D. Thesis. In vitro synthetic transcriptional networks, Jan. 2007.

[50] J. Kim, J. J. Hopfield, and E. Winfree. Neural network computation by in vitro transcriptional circuits, Oct. 2004.

[51] J. Kim, K. S. White, and E. Winfree. Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Molecular Systems Biology*, 2(1):68, Jan. 2006.

[52] J. Kim and E. Winfree. Synthetic in vitro transcriptional oscillators. *Molecular Systems Biology*, 7(465):465, Feb. 2011.

[53] A. Krivoruchko, V. Siewers, and J. Nielsen. Opportunities for yeast metabolic engineering: Lessons from synthetic biology. *Biotechnology Journal*, 6(3):262–76, Mar. 2011.

[54] M. R. Lakin, S. Youssef, L. Cardelli, and A. Phillips. Abstractions for DNA circuit design. *Journal of the Royal Society, Interface / the Royal Society*, 9(68):470–86, Mar. 2012.

[55] M. R. Lakin, S. Youssef, F. Polo, S. Emmott, and A. Phillips. Visual DSD: a design and analysis tool for DNA strand displacement systems. *Bioinformatics*, 27(22):3211–3, Nov. 2011.

[56] C. G. Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1-3):120–149, Oct. 1986.

[57] D. A. Lauffenburger. Cell signaling pathways as control modules: Complexity for simplicity? *Proceedings of the National Academy of Sciences*, 97(10):5031–5033, May 2000.

[58] J. J. Li and W. Tan. A Single DNA Molecule Nanomotor. *Nano Letters*, 2(4):315–318, Apr. 2002.

[59] T. A. Lincoln and G. F. Joyce. Self-sustained replication of an RNA enzyme. *Science*, 323(5918):1229–32, Feb. 2009.

[60] P. L. Luisi, F. Ferri, and P. Stano. Approaches to semi-synthetic minimal cells: a review. *Die Naturwissenschaften*, 93(1):1–13, Jan. 2006.

[61] K. Lund, A. J. Manzo, N. Dabby, N. Michelotti, A. Johnson-Buck, J. Nangreave, S. Taylor, R. Pei, M. N. Stojanovic, N. G. Walter, E. Winfree, and H. Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206–10, May 2010.

[62] C. Mao, W. Sun, Z. Shen, and N. C. Seeman. A nanomechanical device based on the B-Z transition of DNA. *Nature*, 397(6715):144–6, Jan. 1999.

[63] N. R. Markham and M. Zuker. DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Research*, 33(Web Server issue):W577–81, July 2005.

[64] N. R. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybridization. *Methods in Molecular Biology*, 453:3–31, Jan. 2008.

[65] P. Mazzarello. A unifying concept: the history of cell theory. *Nat Cell Biol*, 1(1):E13–E15, May 1999.

[66] H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269(5224):650–656, Aug. 1995.

[67] H. H. McAdams. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, 94(3):814–819, Feb. 1997.

[68] K. Michalodimitrakis and M. Isalan. Engineering prokaryotic gene circuits. *FEMS Microbiology Reviews*, 33(1):27–37, Jan. 2009.

[69] H. I. Miller, A. D. Riggs, and G. N. Gill. Ribonuclease H (Hybrid) in Escherichia coli. IDENTIFICATION AND CHARACTERIZATION. *J. Biol. Chem.*, 248(7):2621–2624, Apr. 1973.

[70] S. Modi, S. M G, D. Goswami, G. D. Gupta, S. Mayor, and Y. Krishnan. A DNA nanomachine that maps spatial and temporal pH changes inside living cells. *Nature Nanotechnology*, 4(5):325–30, May 2009.

[71] K. Montagne, R. Plasson, Y. Sakai, T. Fujii, and Y. Rondelez. Programming an in vitro DNA oscillator using a molecular networking strategy. *Molecular Systems Biology*, 7:466, Feb. 2011.

[72] H. J. Morowitz, B. Heinz, and D. W. Deamer. The chemical logic of a minimum protocell. *Origins of Life and Evolution of the Biosphere*, 18(3):281–287, Sept. 1988.

[73] S. Mukherji and A. van Oudenaarden. Synthetic biology: understanding biological design from synthetic circuits. *Nature Reviews. Genetics*, 10(12):859–71, Dec. 2009.

[74] N. Nandagopal and M. B. Elowitz. Synthetic biology: integrated gene circuits. *Science*, 333(6047):1244–8, Sept. 2011.

[75] J. V. Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

[76] A. Niemz, T. M. Ferguson, and D. S. Boyle. Point-of-care nucleic acid testing for infectious diseases. *Trends in Biotechnology*, 29(5):240–50, May 2011.

[77] V. Noireaux, R. Bar-Ziv, J. Godefroy, H. Salman, and A. Libchaber. Toward an artificial cell based on gene expression in vesicles. *Physical Biology*, 2(3):P1–8, Sept. 2005.

[78] V. Noireaux, R. Bar-Ziv, and A. Libchaber. Principles of cell-free genetic circuit assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 100(22):12672–7, Oct. 2003.

[79] V. Noireaux and A. Libchaber. A vesicle bioreactor as a step toward an artificial cell assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17669–74, Dec. 2004.

[80] K. Oishi and E. Klavins. Biomolecular implementation of linear I/O systems. *IET Systems Biology*, 5(4):252–60, July 2011.

[81] T. Omabegho, R. Sha, and N. C. Seeman. A bipedal DNA Brownian motor with coordinated legs. *Science*, 324(5923):67–71, Apr. 2009.

[82] M. Ptashne. *A genetic switch: Gene control and phage [lambda].* Cell Press, Cambridge, Mass. and Palo Alto, Calif., 1987.

[83] M. Ptashne. *A genetic switch: Gene control and phage [lambda].* Cell Press, Cambridge, Mass. and Palo Alto, Calif., 1987.

[84] P. E. M. Purnick and R. Weiss. The second wave of synthetic biology: from modules to systems. *Nature Reviews. Molecular Cell Biology*, 10(6):410–22, June 2009.

[85] L. Qian and E. Winfree. A simple DNA gate motif for synthesizing large-scale circuits. *Journal of the Royal Society, Interface / the Royal Society*, 8(62):1281–97, Sept. 2011.

[86] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–201, June 2011.

[87] N. Rosenfeld, M. B. Elowitz, and U. Alon. Negative autoregulation speeds the response times of transcription networks. *Journal of Molecular Biology*, 323(5):785–93, Nov. 2002.

[88] N. Rosenfeld, J. W. Young, U. Alon, P. S. Swain, and M. B. Elowitz. Accurate prediction of gene feedback circuit behavior from component properties. *Molecular Systems Biology*, 3:143, Jan. 2007.

[89] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, Mar. 2006.

[90] P. W. K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology*, 2(12):e424, Dec. 2004.

[91] W. C. Ruder, T. Lu, and J. J. Collins. Synthetic biology moving into the clinic. *Science*, 333(6047):1248–52, Sept. 2011.

[92] S. W. Santoro and G. F. Joyce. A general purpose RNA-cleaving DNA enzyme. *Proceedings of the National Academy of Sciences of the United States of America*, 94(9):4262–4266, Apr. 1997.

[93] P. Schwille. Bottom-up synthetic biology: engineering in a tinkerer's world. *Science*, 333(6047):1252–4, Sept. 2011.

[94] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314(5805):1585–8, Dec. 2006.

[95] N. C. Seeman. DNA in a material world. *Nature*, 421(6921):427–31, Jan. 2003.

[96] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–8, May 2002.

[97] W. B. Sherman and N. C. Seeman. A Precisely Controlled DNA Biped Walking Device. *Nano Letters*, 4(7):1203–1207, July 2004.

[98] R. P. Shetty, D. Endy, and T. F. Knight. Engineering BioBrick vectors from BioBrick parts. *Journal of Biological Engineering*, 2:5, Jan. 2008.

[99] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126(35):10834–5, Sept. 2004.

[100] D. Siegal-Gaskins, M. K. Mejia-Guerra, G. D. Smith, and E. Grotewold. Emergence of switch-like behavior in a large family of simple biochemical networks. *PLoS Computational Biology*, 7(5):e1002039, May 2011.

[101] F. C. Simmel and B. Yurke. A DNA-based molecular device switchable between three distinct mechanical states. *Applied Physics Letters*, 80(5):883, Feb. 2002.

[102] C. D. Smolke and P. A. Silver. Informing biological design by integration of systems and synthetic biology. *Cell*, 144(6):855–9, Mar. 2011.

[103] R. V. Solé, A. Munteanu, C. Rodriguez-Caso, and J. Macía. Synthetic protocell biology: from reproduction to computation. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 362(1486):1727–39, Oct. 2007.

[104] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, 107(12):5393–8, Mar. 2010.

[105] D. Sprinzak and M. B. Elowitz. Reconstruction of genetic circuits. *Nature*, 438(7067):443–8, Nov. 2005.

[106] P. Subsoontorn, J. Kim, and E. Winfree. Ensemble Bayesian Analysis of Bistability in a Synthetic Transcriptional Switch. *ACS Synthetic Biology*, 1(8):120626082725007, June 2012.

[107] Y. Tian, Y. He, Y. Chen, P. Yin, and C. Mao. A DNAzyme That Walks Processively and Autonomously along a One-Dimensional Track. *Angewandte Chemie International Edition*, 44(28):4355–4358, 2005.

[108] A. Turberfield, J. Mitchell, B. Yurke, A. Mills, M. Blakey, and F. Simmel. DNA Fuel for Free-Running Nanomachines. *Physical Review Letters*, 90(11), Mar. 2003.

[109] S. Tyagi and F. R. Kramer. Molecular beacons: probes that fluoresce upon hybridization. *Nature Biotechnology*, 14(3):303–8, Mar. 1996.

[110] C. A. Voigt. Genetic parts to program bacteria. *Current Opinion in Biotechnology*, 17(5):548–57, Oct. 2006.

[111] M. Wieland and M. Fussenegger. Engineering molecular circuits using synthetic biology in Mammalian cells. *Annual Review of Chemical and Biomolecular Engineering*, 3:209–34, Jan. 2012.

[112] E. Winfree. Ph.D. Thesis. Algorithmic self-assembly of DNA, May 1998.

[113] E. Winfree. Algorithmic Self-Assembly of DNA: Theoretical Motivations and 2D Assembly Experiments. *Journal of Biomolecular Structure and Dynamics*, 11(2):263–270, 2000.

[114] P. Wu and L. Brand. Resonance energy transfer: methods and applications. *Analytical Biochemistry*, 218(1):1–13, Apr. 1994.

[115] P. Yin, H. M. T. Choi, C. R. Calvert, and N. A. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451(7176):318–22, Jan. 2008.

[116] P. Yin, H. Yan, X. G. Daniell, A. J. Turberfield, and J. H. Reif. A Unidirectional DNA Walker That Moves Autonomously along a Track. *Angewandte Chemie International Edition*, 43(37):4906–4911, 2004.

[117] B. Yurke, A. J. Turberfield, A. P. Mills, F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406(6796):605–8, Aug. 2000.

[118] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, and N. A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–3, Jan. 2011.

[119] J. N. Zadeh, B. R. Wolfe, and N. A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–52, Feb. 2011.

[120] D. Y. Zhang and G. Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–13, Feb. 2011.

[121] D. Y. Zhang, A. J. Turberfield, B. Yurke, and E. Winfree. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, 318(5853):1121–5, Nov. 2007.

[122] D. Y. Zhang and E. Winfree. Dynamic allosteric control of noncovalent DNA catalysis reactions. *Journal of the American Chemical Society*, 130(42):13921–6, Oct. 2008.

[123] D. Y. Zhang and E. Winfree. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–14, Dec. 2009.

[124] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, July 2003.

Appendix A

DESIGNS

A.0.1 Genelet System + Nanomotor

This design modification was an early iteration of the design in Chapter 4, and is missing many of the latter's features. However, this design was intended to drive the DNA nanomotor describe in Chapter 3, instead of the ssDNA probe, as can be seen in the domain-level specifications in Figure A.1. Unfortunately, the features lacking from this design (short input region, no spacer, high sequence similarity between genelet and nanomotor, etc.) means that this design should be redone before being tested.

A.0.2 Genelet System + Malachite Green Aptamer

This design modification includes a new feature, as can be seen in the domain-level specifications in Figure A.2. This design includes a malachite green aptamer sequence on the template, so that the aptamer is transcribed at the $3'$ end of the RNA output. The malachite green apatmer fluoresces when bound to malachite green in solution, and so reports on $[r]$ directly. Unfortunately, the malachite green reacts with the mineral oil that prevents evaporation from wells.

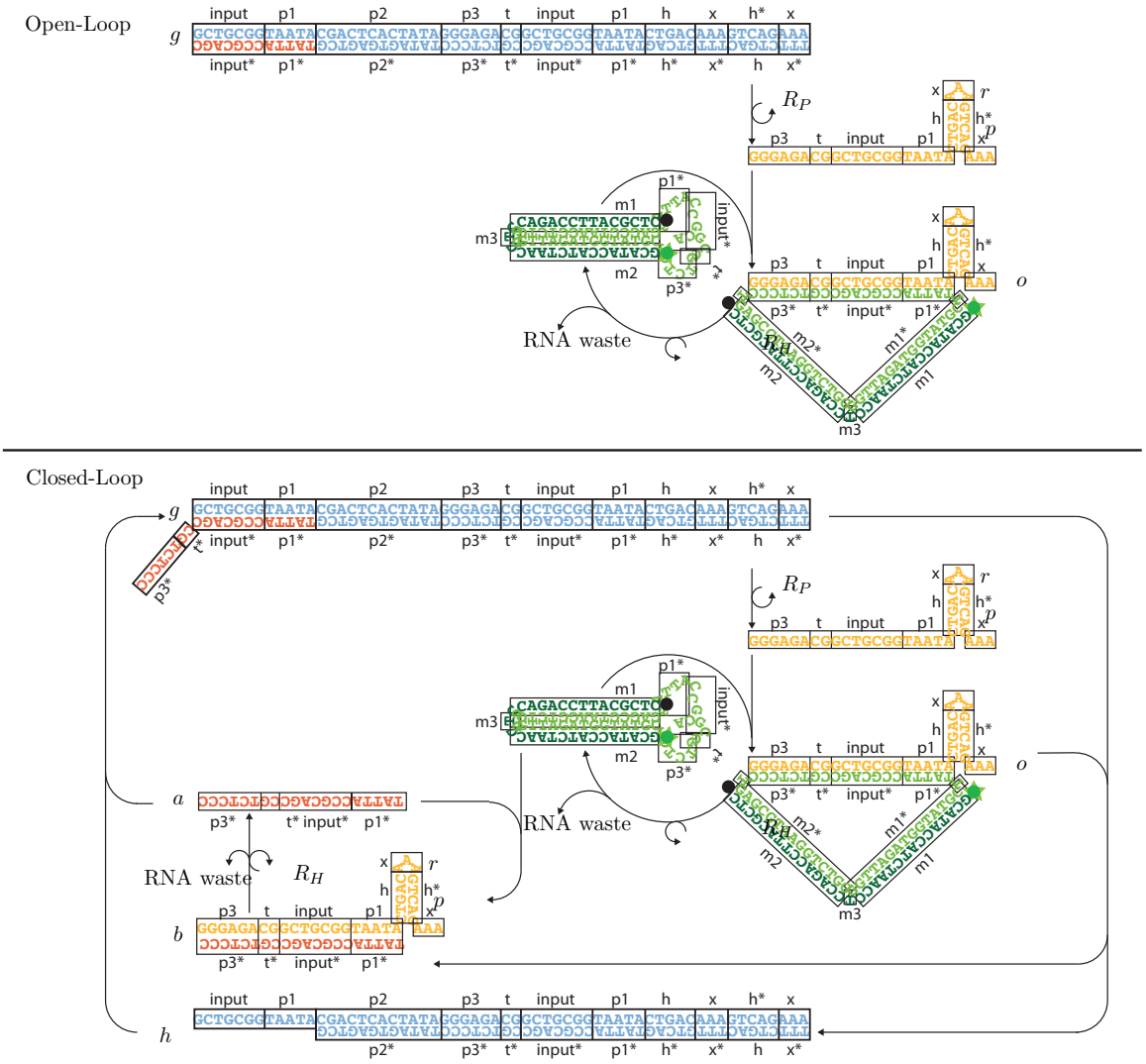

Figure A.1: Domain-level specification for open- and closed-loop genelet systems that drive the DNA nanomotor described in Chapter 3. Domains are indicated by black boxes, with domain names above and/or below as appropriate. The symbol * indicates complementarity between domains with the same name.

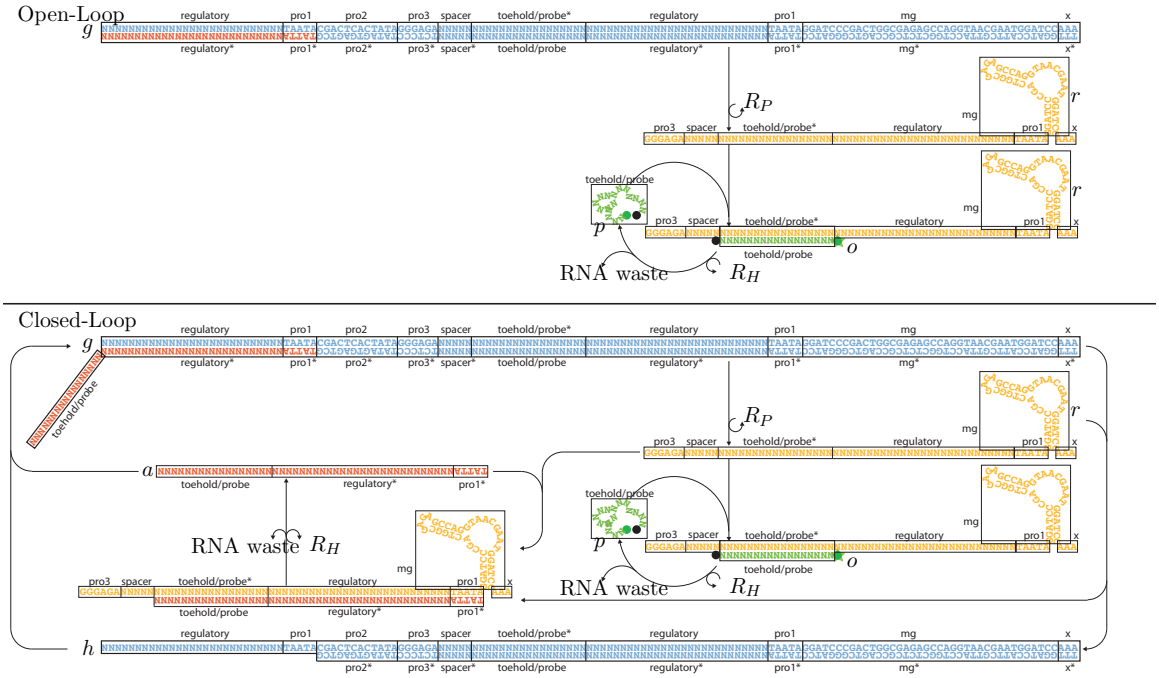

Figure A.2: Domain-level specification for open- and closed-loop genelet systems that use a malachite green aptamer to report the concentration of r. Domains are indicated by black boxes, with domain names above and/or below as appropriate. The symbol * indicates complementarity between domains with the same name. The domains with pre-defined sequences are the promoter region, `pro1`, `pro2`, and `pro3`; the probe, which also defines the activator toehold region, `toehold/probe`; and the malachite green aptamer region `mg`. The domains with undefined sequences are the genelet regulatory region `regulatory`; the inert region `spacer`, which ensures abortive transcripts are inert; and the hairpin stem regions `hairpin` and `hairpin`*.

Appendix B

SBML MODEL

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version3" level
="2" version="3">
  <model id="net1">
    <listOfUnitDefinitions>
      <unitDefinition id="substance">
        <listOfUnits>
          <unit kind="mole" scale="-9"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="time">
        <listOfUnits>
          <unit kind="second" multiplier="3600"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="nM" name="nM">
        <listOfUnits>
          <unit kind="mole" scale="-9"/>
          <unit kind="litre" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="per_nM_per_h" name="/nM/s">
        <listOfUnits>
          <unit kind="litre"/>
```

```
24            <unit kind="mole" scale="-9" exponent="-1"/>
25            <unit kind="second" multiplier="3600" exponent
="-1"/>
26          </listOfUnits>
27        </unitDefinition>
28        <unitDefinition id="per_nM" name="/nM">
29          <listOfUnits>
30            <unit kind="litre"/>
31            <unit kind="mole" scale="-9" exponent="-1"/>
32          </listOfUnits>
33        </unitDefinition>
34        <unitDefinition id="per_h" name="/h">
35          <listOfUnits>
36            <unit kind="second" multiplier="3600" exponent
="-1"/>
37          </listOfUnits>
38        </unitDefinition>
39      </listOfUnitDefinitions>
40      <listOfCompartments>
41        <compartment id="Well" name="Well" size="1"
spatialDimensions="3"/>
42      </listOfCompartments>
43      <listOfSpecies>
44        <species id="g" compartment="Well" initialConcentration
="0"/>
45        <species id="s" compartment="Well" initialConcentration
="0"/>
46        <species id="a" compartment="Well" initialConcentration
="0"/>
47        <species id="gs" compartment="Well" initialConcentration
="0"/>
```

```
48        <species id="ga" compartment="Well" initialConcentration
  ="0"/>
49        <species id="r" compartment="Well" initialConcentration
  ="0"/>
50        <species id="ar" compartment="Well" initialConcentration
  ="0"/>
51        <species id="p" compartment="Well" initialConcentration
  ="100"/>
52        <species id="pr" compartment="Well" initialConcentration
  ="0"/>
53        <species id="y" compartment="Well"/>
54      </listOfSpecies>
55      <listOfParameters>
56        <parameter id="kr1" value="0.0121644" units="per_h"/>
57        <parameter id="kr2" value="8.00839" units="per_h"/>
58        <parameter id="kd1" value="82.2627" units="per_h"/>
59        <parameter id="kd2" value="12.6569" units="per_h"/>
60        <parameter id="ka" value="0.0000231405" units="
  per_nM_per_h"/>
61        <parameter id="kf" value="0.000163101" units="
  per_nM_per_h"/>
62        <parameter id="ko" value="0.00159884" units="
  per_nM_per_h"/>
63        <parameter id="kp" value="27274.7" units="per_nM_per_h
  "/>
64        <parameter id="ky1" value="44.8071" units="per_nM"/>
65        <parameter id="ky2" value="0.0" units="per_nM"/>
66      </listOfParameters>
67      <listOfRules>
68        <assignmentRule variable="y">
69          <math xmlns="http://www.w3.org/1998/Math/MathML">
70            <apply>
```

```
71              <plus/>
72              <apply>
73                <times/>
74                <ci> ky1 </ci>
75                <ci> pr </ci>
76                <ci> Well </ci>
77              </apply>
78              <apply>
79                <times/>
80                <ci> ky2 </ci>
81                <apply>
82                  <plus/>
83                  <ci> pr </ci>
84                  <ci> p </ci>
85                </apply>
86                <ci> Well </ci>
87              </apply>
88            </apply>
89          </math>
90        </assignmentRule>
91      </listOfRules>
92      <listOfReactions>
93        <reaction id="R1">
94          <listOfProducts>
95            <speciesReference species="r"/>
96          </listOfProducts>
97          <listOfModifiers>
98            <modifierSpeciesReference species="g"/>
99          </listOfModifiers>
100         <kineticLaw>
```

```
101           <math xmlns="http://www.w3.org/1998/Math/MathML">
102             <apply>
103               <times/>
104               <ci> Well </ci>
105               <ci> kr2 </ci>
106               <ci> g </ci>
107             </apply>
108           </math>
109         </kineticLaw>
110       </reaction>
111       <reaction id="R2">
112         <listOfProducts>
113           <speciesReference species="r"/>
114         </listOfProducts>
115         <listOfModifiers>
116           <modifierSpeciesReference species="gs"/>
117         </listOfModifiers>
118         <kineticLaw>
119           <math xmlns="http://www.w3.org/1998/Math/MathML">
120             <apply>
121               <times/>
122               <ci> Well </ci>
123               <ci> kr1 </ci>
124               <ci> gs </ci>
125             </apply>
126           </math>
127         </kineticLaw>
128       </reaction>
129       <reaction id="R3">
130         <listOfProducts>
```

```
131           <speciesReference species="r"/>
132         </listOfProducts>
133         <listOfModifiers>
134           <modifierSpeciesReference species="ga"/>
135         </listOfModifiers>
136         <kineticLaw>
137           <math xmlns="http://www.w3.org/1998/Math/MathML">
138             <apply>
139               <times/>
140               <ci> Well </ci>
141               <ci> kr1 </ci>
142               <ci> ga </ci>
143             </apply>
144           </math>
145         </kineticLaw>
146       </reaction>
147       <reaction id="R4">
148         <listOfReactants>
149           <speciesReference species="p"/>
150           <speciesReference species="r"/>
151         </listOfReactants>
152         <listOfProducts>
153           <speciesReference species="pr"/>
154         </listOfProducts>
155         <kineticLaw>
156           <math xmlns="http://www.w3.org/1998/Math/MathML">
157             <apply>
158               <times/>
159               <ci> Well </ci>
160               <ci> kp </ci>
```

```
161               <ci> p </ci>
162               <ci> r </ci>
163             </apply>
164           </math>
165         </kineticLaw>
166       </reaction>
167       <reaction id="R5">
168         <listOfReactants>
169           <speciesReference species="pr"/>
170         </listOfReactants>
171         <listOfProducts>
172           <speciesReference species="p"/>
173         </listOfProducts>
174         <kineticLaw>
175           <math xmlns="http://www.w3.org/1998/Math/MathML">
176             <apply>
177               <times/>
178               <ci> Well </ci>
179               <ci> kd1 </ci>
180               <ci> pr </ci>
181             </apply>
182           </math>
183         </kineticLaw>
184       </reaction>
185       <reaction id="R6">
186         <listOfReactants>
187           <speciesReference species="ga"/>
188           <speciesReference species="r"/>
189         </listOfReactants>
190         <listOfProducts>
```

```
191           <speciesReference species="ar"/>
192           <speciesReference species="g"/>
193         </listOfProducts>
194         <kineticLaw>
195           <math xmlns="http://www.w3.org/1998/Math/MathML">
196             <apply>
197               <times/>
198               <ci> Well </ci>
199               <ci> kf </ci>
200               <ci> ga </ci>
201               <ci> r </ci>
202             </apply>
203           </math>
204         </kineticLaw>
205       </reaction>
206       <reaction id="R7">
207         <listOfReactants>
208           <speciesReference species="a"/>
209           <speciesReference species="r"/>
210         </listOfReactants>
211         <listOfProducts>
212           <speciesReference species="ar"/>
213         </listOfProducts>
214         <kineticLaw>
215           <math xmlns="http://www.w3.org/1998/Math/MathML">
216             <apply>
217               <times/>
218               <ci> Well </ci>
219               <ci> ka </ci>
220               <ci> a </ci>
```

```
221             <ci> r </ci>
222           </apply>
223         </math>
224       </kineticLaw>
225     </reaction>
226     <reaction id="R8">
227       <listOfReactants>
228         <speciesReference species="g"/>
229         <speciesReference species="a"/>
230       </listOfReactants>
231       <listOfProducts>
232         <speciesReference species="ga"/>
233       </listOfProducts>
234       <kineticLaw>
235         <math xmlns="http://www.w3.org/1998/Math/MathML">
236           <apply>
237             <times/>
238             <ci> Well </ci>
239             <ci> ko </ci>
240             <ci> g </ci>
241             <ci> a </ci>
242           </apply>
243         </math>
244       </kineticLaw>
245     </reaction>
246     <reaction id="R9">
247       <listOfReactants>
248         <speciesReference species="ar"/>
249       </listOfReactants>
250       <listOfProducts>
```

```
251           <speciesReference species="a"/>
252         </listOfProducts>
253         <kineticLaw>
254           <math xmlns="http://www.w3.org/1998/Math/MathML">
255             <apply>
256               <times/>
257               <ci> Well </ci>
258               <ci> kd2 </ci>
259               <ci> ar </ci>
260             </apply>
261           </math>
262         </kineticLaw>
263       </reaction>
264       <reaction id="R10">
265         <listOfReactants>
266           <speciesReference species="g"/>
267           <speciesReference species="s"/>
268         </listOfReactants>
269         <listOfProducts>
270           <speciesReference species="gs"/>
271         </listOfProducts>
272         <kineticLaw>
273           <math xmlns="http://www.w3.org/1998/Math/MathML">
274             <apply>
275               <times/>
276               <ci> Well </ci>
277               <ci> ko </ci>
278               <ci> g </ci>
279               <ci> s </ci>
280             </apply>
```

```
281           </math>
282         </kineticLaw>
283       </reaction>
284     </listOfReactions>
285     <listOfEvents>
286       <event>
287         <trigger>
288           <math xmlns="http://www.w3.org/1998/Math/MathML">
289             <apply>
290                <gt/>
291               <csymbol encoding="text" definitionURL="http://
www.sbml.org/sbml/symbols/time"> time
292               </csymbol>
293               <cn> 1.2 </cn>
294             </apply>
295           </math>
296         </trigger>
297         <listOfEventAssignments>
298           <eventAssignment variable="p">
299             <math xmlns="http://www.w3.org/1998/Math/MathML">
300               <apply>
301                 <plus/>
302                 <ci> p </ci>
303                 <cn> 100 </cn>
304               </apply>
305             </math>
306           </eventAssignment>
307         </listOfEventAssignments>
308       </event>
309       <event>
```

```
310         <trigger>
311           <math xmlns="http://www.w3.org/1998/Math/MathML">
312             <apply>
313               <gt/>
314               <csymbol encoding="text" definitionURL="http://
    www.sbml.org/sbml/symbols/time"> time
315               </csymbol>
316               <cn> 2.38333 </cn>
317             </apply>
318           </math>
319         </trigger>
320         <listOfEventAssignments>
321           <eventAssignment variable="p">
322             <math xmlns="http://www.w3.org/1998/Math/MathML">
323               <apply>
324                 <plus/>
325                 <ci> p </ci>
326                 <cn> 100 </cn>
327               </apply>
328             </math>
329           </eventAssignment>
330         </listOfEventAssignments>
331       </event>
332       <event>
333         <trigger>
334           <math xmlns="http://www.w3.org/1998/Math/MathML">
335             <apply>
336               <gt/>
337               <csymbol encoding="text" definitionURL="http://
    www.sbml.org/sbml/symbols/time"> time
```

```
338                 </csymbol>
339                 <cn> 3.56667 </cn>
340               </apply>
341             </math>
342           </trigger>
343            <listOfEventAssignments>
344             <eventAssignment variable="p">
345               <math xmlns="http://www.w3.org/1998/Math/MathML">
346                 <apply>
347                   <plus/>
348                   <ci> p </ci>
349                   <cn> 100 </cn>
350                 </apply>
351               </math>
352             </eventAssignment>
353           </listOfEventAssignments>
354         </event>
355       </listOfEvents>
356     </model>
357 </sbml>
```

Appendix C

PROTOCOLS

C.1 Experiments

System trajectories, in which the genelet systems responded to two exogenous additions of $[p]^{tot}$ under various initial conditions, were observed via fluorescence activity of p in the presence of g; either s or a; both R_P and R_H; and the necessary *in vitro* transcription buffer conditions. We recently generated three sets of data by varying s^{tot} and a^{tot} under two sets of conditions. The general protocol used for collecting this data follows.

C.1.1 Data collection protocol

1. Combine in a 1.5-mL reaction tube (label "buffer mix"), at a scale proportional to the number of experiments (where the total volume of experiments is 100 μL, see Table 6.1 for final reagent concentrations in each dataset), in order:
    - molecular grade H_2O,
    - 10x Transcription Buffer (NEB),
    - 80 mM rNTPs (NEB),
    - 100 mM $MgCl_2$ and
    - 25 μM g (previously annealed in H_2O and 10x Transcription Buffer).
2. Vortex "buffer mix" for 1 minute.
3. Distribute in appropriate volume to wells in a Corning 3880 microplate.
4. To each well containing experiments:
    (a) Add 1 μL of either s or a from stock (100$\times$ the desired s^{tot} or a^{tot}).

(b) Mix mechanically using the tip of the pipette, with rapidity.

(c) Coat with 50 μL of mineral oil.

5. Cover plate with a plastic lid and place in a Synergy HT plate reader (BioTek).

6. Set plate reader to collect raw fluorescence data

   - from bottom of each well,
   - once a minute,
   - with sensitivity 56,
   - using 575/15 excitation and 620/15 emission wavelength/bandpass filters,
   - at 37°C.

7. Program plate reader to

   (a) collect data for 30 minutes,

   (b) eject the plate for additions,

   (c) collect data for 8 hours,

   (d) eject the plate for additions again, and

   (e) collect data for another 24 hours.

8. Start plate reader program at 7(a). Meanwhile, combine in a 0.5-mL reaction tube (label "enzyme mix") at a scale proportional to the number of experiments, in order:

   - molecular grade H_2O,
   - 10 μM p,
   - 2 U/μL TIPP (thermostable inorganic pyrophosphatase, NEB)
   - 10 U/μL R_H (EpiBio)
   - 2 U/μL R_P (EpiBio)

9. Vortex “enzyme mix” for 1 minute.

10. At 7(b), remove plate from plate reader tray and incubate on a heat block at 37°C.

11. To each well containing experiments:

    (a) Add appropriate volume of ”enzyme mix”.

    (b) Mix mechanically using the tip of the pipette, with rapidity.

12. Centrifuge plate at 1000 × g for 1 second, then replace plate in plate reader and start 7(c).

13. At 7(d), remove plate from plate reader tray and incubate on a heat block at 37°C.

14. To each well containing experiments:

    (a) Add 1 μL of p from stock.

    (b) Mix mechanically using the tip of the pipette, with rapidity.

15. Centrifuge plate at 1000 × g for 1 second, then replace in plate reader and start 7(e).

C.1.2 Data calibration

System trajectory data was calibrated using a function that relates raw fluorescence values to concentrations of open probe, i.e.,

$$\text{fluorescence} = a_0 + a_1[\text{closed probe}] + a_2[\text{open probe}],$$

or equivalently,

$$[\text{open probe}] = \frac{\text{fluorescence} - a_0 - a_1\, p^{tot}}{a_2 - a_1}$$

since $p^{tot} = [\text{closed probe}] + [\text{open probe}]$.

The parameters of this function were estimated empirically in experiments in which probe was titrated into standard buffer conditions with and without the probe’s DNA complement, p^*. The protocol for these experiments was as follows.

1. Combine in a 1.5-mL reaction tube (label "buffer mix"), in order:

    - 340.3 μL of molecular grade H_2O,
    - 36.9 μL of 10x Transcription Buffer (NEB),
    - 20.5 μL of 80 mM rNTPs (NEB),
    - 8.2 μL of 100 mM $MgCl_2$.

    (a) Vortex "buffer mix" for 1 minute.

    (b) Distribute 99 μL from "buffer mix" to each of four (4) wells in a Corning 3880 microplate.

    (c) Coat with 50 μL of mineral oil.

2. Combine in a 1.5-mL reaction tube (label "complement mix"), in order:

    - 336.2 μL of molecular grade H_2O,
    - 36.9 μL of 10x Transcription Buffer (NEB),
    - 20.5 μL of 80 mM rNTPs (NEB),
    - 8.2 μL of 100 mM $MgCl_2$,
    - 4.1 μL of 100 μM p^*.

    (a) Vortex "complement mix" for 1 minute.

    (b) Distribute 99 μL from "complement mix" to each of four (4) wells in a Corning 3880 microplate.

    (c) Coat with 50 μL of mineral oil.

3. Cover plate with a plastic lid and place in a Synergy HT plate reader (BioTek).

4. Set plate reader to collect raw fluorescence data

    - from bottom of each well,

Table C.1: Calibration test reagents

iteration	1	2	3	4	5	6	7	8	9
stock (μM)	0.5	0.5	0.5	0.5	0.5	0.5	10	10	10
volume (μL)	1	1	1	1	1	1	0.25	0.5	1
final (nM)	5	9.95	14.85	19.71	24.51	29.28	53.03	100.311	193.99

- once a minute,
- with sensitivity 56,
- using 575/15 excitation and 620/15 emission wavelength/bandpass filters,
- at 37°C.

5. Program plate reader to
   (a) collect data for 10 minutes,
   (b) eject the plate for additions,
   (c) repeat (a) and (b).
6. Start plate reader program at 7(a).
7. At each iteration of 7(b), remove plate from plate reader tray and incubate on a heat block at 37°C.
   (a) To each well containing experiments add p according to this table.
   (b) Mix mechanically using the tip of the pipette, with rapidity.
8. Centrifuge plate at 1000 × g for 1 second, then replace plate in plate reader and start 7(c).

VITA

Josh Bishop is from Louisville, Kentucky, where the majority of his family still reside. He has lived in Seattle, Washington, which is where he met his wife, for just over a decade. He earned a Bachelor of Science degree in Computer Engineering from the University of Illinois in 2001. He also earned a Masters degree in Electrical Engineering at the University of Washington after completing coursework in Control Systems and Robotics, before entering the doctoral program of the same department.