CORRECTING ERRORS DUE TO SPECIES CORRELATIONS IN THE MARGINAL PROBABILITY DENSITY EVOLUTION ALGORITHM

by

Abiezer Tejeda

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

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Chris Winstead Major Professor	Dr. Reyhan Baktur Committee Member
Dr. Charles D. Miller Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013

Abstract

Correcting Errors Due to Species Correlations in the Marginal Probability Density Evolution Algorithm

by

Abiezer Tejeda, Master of Science
Utah State University, 2013

Major Professor: Dr. Chris Winstead
Department: Electrical and Computer Engineering

Synthetic biology is an emerging field that integrates and applies engineering design methods to biological systems. Its aim is to make biology an "engineerable" science. Over the years, biologists and engineers alike have abstracted biological systems into functional models that behave similarly to electric circuits, thus the creation of the subfield of genetic circuits. Mathematical models have been devised to simulate the behavior of genetic circuits *in silico*. Most models can be classified into *deterministic* and *stochastic* models. The work in this dissertation is for stochastic models.

Although ordinary differential equation (ODE) models are generally amenable to simulate genetic circuits, they wrongly assume that a system's chemical species vary continuously and deterministically, thus making erroneous predictions when applied to highly stochastic systems. Stochastic methods have been created to take into account the variability, unpredictability, and discrete nature of molecular populations. The most popular stochastic method is the *stochastic simulation algorithm* (SSA). These methods provide a single path of the overall pool of possible system's behavior. A common practice is to take several independent SSA simulations and take the average of the aggregate. This approach can perform well in low noise systems. However, it produces incorrect results when applied to networks that can take multiple modes or that are highly stochastic.

Incremental SSA or iSSA is a set of algorithms that have been created to obtain aggregate information from multiple SSA runs. The marginal probability density evolution (MPDE) algorithm is a subset of iSSA which seeks to reveal the most likely "qualitative" behavior of a genetic circuit by providing a marginal probability function or statistical envelope for every species in the system, under the appropriate conditions. MPDE assumes that species are statistically independent given the rest of the system. This assumption is satisfied by some systems. However, most of the interesting biological systems, both synthetic and in nature, have correlated species forming conservation laws. Species correlation imposes constraints in the system that are broken by MPDE. This work seeks to devise a mathematical method and algorithm to correct conservation constraints errors in MPDE. Furthermore, it aims to identify these constraints *a priori* and efficiently deliver a trustworthy result faithful to the true behavior of the system.

(90 pages)

Public Abstract

Correcting Errors Due to Species Correlations in the Marginal Probability Density Evolution Algorithm

by

Abiezer Tejeda, Master of Science
Utah State University, 2013

Major Professor: Dr. Chris Winstead
Department: Electrical and Computer Engineering

Synthetic biology is a fairly new science which is concerned about making biology easy to engineer. It combines concepts from engineering and biology to try to design, simulate, predict, and build synthetic living systems. In order to make this possible, computer-aided design (CAD) tools are needed by designers to help design and simulate the system before it is built in the lab. Over the years several distinct mathematical methods have been created to model and simulate the behavior of biochemical networks. Ordinary differential equations are among the oldest of these methods. Although differential equations have been successfully applied to simple systems, they make erroneous assumptions that fail to produce the correct results when applied to highly variable systems. For this reason, *stochastic* models have been created to take into account the randomness or variability of biochemical networks. The most popular of the stochastic models is commonly known as Stochastic Simulation Algorithm (SSA). SSA can show a single path or result from the many possible solutions or behaviors the system can take. In order to figure out what is the most likely solution, researchers take multiple independent runs of SSA and take the average. This technique can yield correct results only for systems that are not highly stochastic but the technique fails when applied to highly random systems. For this reason the *marginal* *probability density evolution* (MPDE) algorithm was created to deliver a trustworthy and robust result of the expected behavior of the system. Although MPDE works for many simple models, it also has its limitations. MPDE fails when applied to systems that exhibit conservation constraints. Conservation constraints are mathematical laws that are imposed to the system. For instance, a conservation law might be that the sum of the molecules of two biological species must remain constant at all times. This work seeks to refine MPDE by creating an algorithm to take into account conservation laws present in the genetic circuit model. A refined version of MPDE will be able to simulate a wider array of interesting biological models while producing correct and robust results close to those observed in the lab or in nature.

To my parents, sister, friends, and to Dr. Chris Winstead...

Acknowledgments

I would like to express my gratitude to Dr. Chris Winstead for his mentorship, guidance, and support throughout this project. Special thanks go to Eduardo Monzon for helping with simulations, advice, and productive conversations on several topics on the project. I would also like to thank Curtis Kendall and Chris Myers at the University of Utah for sharing their experience on iBioSim, a tool used to produce many of the results shown in this thesis. There are other people that helped hone this work and to whom I am greatly thankful: Francisco Santos for creating and polishing some of the figures, as well as Miguel Leonardo, Andres Contreras, Manuel Diaz, Ricardo Estevez, Gopalakrishnan Sundararjan, David Toribio, and also Mary Lee Anderson for proofreading and correcting this work. Finally, I would like to express my gratitude to my committee for taking the time to review and evaluate this work as well as sharing their expertise.

Abiezer Tejeda

Contents

List of Tables

Table Page

List of Figures

Figure Page

Acronyms

CCK	Classical Chemical Kinetis
DNA	Deoxiribonucleic Acid
iSSA	Incremental Simulation Stochastic Algorithm
IC	Integrated Circuit
mRNA	Messenger RNA
tRNA	Transfer RNA
MP	Mean Path
MPDE	Marginal Probability Density Evolution
ODE	Ordinary Differential Equation
PC	Principal Component
PCA	Principal Component Analysis
RNA	Ribonucleic Acic
RNAP	RNA Polymerase
SSA	Simulation Stochastic Algorithm
SVD	Singular Value Decomposition
TF	Transcription Factor

Chapter 1

Introduction

Synthetic biology is an emerging science concerned about new ways to engineer biological systems. The subfield of genetic circuits consists of methods and tools for designing functional behavior in organisms by inserting exogenous genetic instructions. One major area of research for genetic circuits is to analyze and predict the behavior of synthetic gene networks by means of computational tools [1]. However, the randomness of these circuits makes *in silico* analysis cumbersome [2]. Moreover, due to complex protein interactions and stochastic events, it is difficult to establish truly modular functional models for genetic parts.

Mathematical models have been created to characterize, predict, and modify the behavior of genetically engineered networks. Chemical reaction networks can be transformed into a set of first order differential equations (ODEs). Although ODE models are generally amenable to modular descriptions, they wrongly assume that a system's chemical species vary deterministically and continuously, which often results in erroneous states [3]. Hence, ODE models can make incorrect predictions when applied to highly stochastic systems, thus requiring stochastic analysis for accurate and robust design of genetic circuits.

To arrive at a modular approach to stochastic genetic circuits, some researchers propose using probability-transfer models [4]. Probabilistic models show some promise for modular synthesis strategies in the forward design of genetic circuits. Nguyen *et al.* recently demonstrated a modular probabilistic approach for synthesizing a “quorum trigger” circuit [5]. In this example, the probabilistic model provided three main benefits: (1) Intuitive abstract behavioral models of the circuit's genetic components; (2) A coherent procedure for forward-design based on modular logic parts; and (3) A framework for estimating the reliability of the synthesized function.

There are two basic questions researchers ask when analyzing biochemical reaction systems. First, what is the typical behavior of the system? Second, how robust or how confident can one be of that behavior? These are very important questions when designing synthetic biological networks. The designer is interested in matching the intended behavior with the actual system's behavior. Popular stochastic simulation algorithms (SSA), such as Gillespie's SSA [6, 7] and τ-leaping [7, 8], provide a single path of the possible system's behavior. In highly stochastic genetic circuits this typical behavior may be concealed by transient "noise."

It is a common practice to execute several independent simulation runs and compute the average over all time to understand the typical behavior of the system. An envelope is then calculated in the form $\bar{\mathbf{x}} \pm \sigma$, where $\bar{\mathbf{x}}$ and σ are the average and standard deviation vectors, respectively, computed over N SSA runs. The average, $\bar{\mathbf{x}}$, is considered to be the system's typical behavior, while the standard deviation, σ, is a confidence envelope indicating how the system can deviate from the expected behavior.

The method of averaging performs well in low noise systems. However, most interesting genetic circuits are highly stochastic and/or multi-stable, for which this method produces an incorrect result. For instance, consider the lambda phage bistable switch [9, 10]. This system can take two possible states. Assuming half of the runs fall into state 1 and half of the runs fall into state 2, taking the average over all time would yield a fictitious "middle" state that conceals the true behavior of the system.

In order to obtain aggregate information from multiple SSA runs, the iSSA method was proposed by Winstead *et al.* [11]. In iSSA, N independent SSA runs are simulated over a short time interval. Statistics are then gathered at the end of the time interval. A new state is computed from those statistics and the algorithm is repeated for another time increment until the simulation time is reached. iSSA can choose one of several methods to compute the statistics. This work focuses in only one of the iSSA methods known as iSSA-MPDE or MPDE for short, meaning *Marginal Probability Density Evolution* [11]. MPDE allows the use of different SSA methods such as τ-leaping, Gibson-Brucks, etc. under certain conditions.

The aim of MPDE is to provide an alternative approach that can reveal the statistical envelope for every species in the system, under the appropriate conditions. MPDE approximates chemical species as a set *independent Gaussian random variables.* At the start of each SSA run, the initial molecule count of each species is computed using each species' marginal Gaussian distribution. When all N SSA runs are terminated, the marginal distributions are estimated by computing the mean and variance of each species. MPDE follows the system's deviation as it evolves over time providing a confidence level of the system's stochasticity and robustness.

In its original presentation, MPDE relied on the assumption that, during a brief time-interval, all species variations are pair-wise conditionally independent, given the system's total state. This assumption is not always accurate. In fact, some variables may be highly dependent on each other, which may completely invalidate the simulation results. Previous accounts of the MPDE method offered no means of testing for dependencies. This work proposes a technique to identify some useful procedures for testing and resolving variable dependencies in MPDE simulations.

The new approach introduces a Linear Gaussian Network (LGN) approximation during brief time intervals of the stochastic simulation. A LGN considers the system as linear for a very short interval of time as well as Gaussian distributed species. This approach can be used to establish statistical independence among species in order to create modular models. The approach can also be used to verify independence assumptions in circuits that are synthesized from modular models.

In most cases, highly correlated species can be identified *a priori* by calculating conservation constraints in the reaction network model. For the remaining independent variables, it is helpful to approximate some, or all, of the system's species as Gaussian-distributed variables. Then, during a brief interval of time, the system may be treated as a LGN. Variable dependencies appear as significant non-zero entries in an LGN's information matrix, which is computed as the pseudo-inverse of the covariance matrix. The information matrix can be computed periodically during simulation, and can be used to spot dependencies in the reaction system (and hence to flag violations of modular independence assumptions).

In addition to the information matrix, dependencies among species of a reaction network appear as linear relationships in the row space of the network stoichiometry matrix [12]. Stoichiometric analysis can allow us to identify conservation relations in the reaction model even before simulation of the genetic circuit is initiated [13,14]. We propose the combination of conservation constraints identification, as presented by Sauro and Ingalls [12], with MPDE in such a way that conservation constraint failures are resolved.

Resolving conservation constraints in MPDE will make the algorithm more suitable for simulating a wider array of interesting biological systems while providing true sample trajectories and meaningful statistical analysis tools to the biological designer.

1.1 Contributions of Thesis

The first contribution of this thesis is the application of a systematic method to identify correlated species within a biochemical model. Although this mathematical methods have existed for hundreds of years, it is the first time they are used in iSSA-MPDE. The second contribution is the creation of an algorithm that effectively partitions a biochemical network into independent and dependent species. This partition is used to keep conservation constraints intact during simulation. In addition to conservation constraints, other subtle types of correlations may appear in a reaction network. A covariance matrix or information matrix can be used to spot those correlations and Principal Component Analysis (PCA) can be used to correct those. Finally, the major contribution of this work is a refined, more robust overall MPDE algorithm that can operate on a wider array of interesting biological systems.

These contributions can be summarized as follows:

- Conservation constraints resolution in MPDE;
- Systematic method to pinpoint correlated species withing a chemical network;

- Run time verification of other subtle types of correlations by using a covariance or information matrix within MPDE.

1.2 Overview of Thesis

This thesis is organized as follows: Chapter 2 presents the reader with the foundations and building blocks necessary to have a rudimentary understanding of synthetic biology and genetic circuits modeling. This chapter introduces basic theory of chemical reactions, cell composition and the central dogma, a brief introduction to synthetic biolgy, and modeling and construction of genetic circuits. Chapter 3 explains the most used mathematical methods to simulate genetic circuits such as ordinary differential equations (ODEs) and some stochastic methods as well. Chapter 4 explains the new stochastic simulation method called *marginal probability density evolution* (MPDE) and shows how species correlation (variable dependencies) can be resolved using conservation constraints analysis. Chapter 5 presents the results found by this investigation while Chapter 6 finalizes with a discussion of the results and methods used in this work.

Chapter 2

Background

This chapter introduces the basic concepts in biology and biochemistry required to understand synthetic biology and genetic circuits. It is organized as follows: Section 2.1 gives an overview of chemical reactions. Section 2.2 presents the basics of synthetic biology, while genetic circuits are described in Section 2.3. Section 2.4 presents an overview genetic circuits modeling techniques and, finally, Section 2.5 briefly describes the process of manufacturing genetic circuits and the registry of standard biological parts (BioBricks).

2.1 Chemical Reactions

A *chemical reaction* is the process by which two or more substances, called *reactants*, are converted into one or more different substances, called the *products*. At the most basic level, chemical reactions combine atoms, the basic building blocks for all matter (living or not), to form molecules and these molecules can also be combined to form more complex compounds. There are three different types of atomic bonds: *covalent, ionic,* and *hydrogen bonds*.

Chemical equations are used to represent chemical reactions mathematically or graphically. For instance, Equation (2.1) is a simple chemical equation for the formation of water. H_2, O_2, and H_2O are referred to as the chemical species, where H_2 and O_2 are the *reactants* or substrates and H_2O is the *product*. The subscripts in H_2 and O_2 indicate that the hydrogen and oxygen molecules are composed of two atoms of the same type. The coefficients in front of the hydrogen (H_2), oxygen (O_2), and water (H_2O) molecules are referred to as the *stoichiometry* of the reaction, indicating that two hydrogen molecules or dimers and one oxygen dimer are used to produce two water molecules. Due to the conservation of matter, the numbers along each side of the equation must be equal.

$$2H_2 + O_2 \overset{k}{\rightarrow} 2H_2O \tag{2.1}$$

The k above the arrow in Equation (2.1) is called the *rate constant*. This value is a measure of how fast this reaction can occur. In practice this value is very difficult to determine exactly for biochemical reactions. Nonetheless, it is used in several of the modeling techniques that will be addressed in this work. In chemistry, the *law of mass action* explains and predicts behaviors in the dynamic equilibrium of chemical reactions. One aspect of this law concerns the kinetics of biochemical reactions, also called rate equations. This law states that the rate of a reaction is determined by the rate constant and the molecule concentrations of reactants raised to the power of their stoichiometry [3]. The rate equation corresponding to Equation (2.1) is shown in Equation (2.2), where $[H_2]$ and $[O_2]$ refer to the concentrations of hydrogen and oxygen, respectively. The 2 in front of the k means that two molecules of water are formed for every reaction that occurs.

$$\frac{d[H_2O]}{dt} = 2k[H_2]^2[O_2] \tag{2.2}$$

Chemical reactions can be divided into different groups according to the type of the reaction. In this work only the most simple form of chemical reactions is considered: *elementary reactions*. An elementary reaction is the smallest division into which a chemical reaction can be decomposed. There are no intermediate products in an elementary reaction. Usually one or two molecules are involved, since the probability for three or more molecules colliding at the same time is very unlikely. The most important types of elementary reactions are unimolecular and bimolecular reactions. Only one molecule is involved in unimolecular reactions while only two molecules participate in bimolecular reactions. These reactions can be either *synthesis*, where two simple reactants combine to form a more complex molecule, or *decomposition*, where a complex substance breaks down into its component molecules. The general form of *synthesis* and *decomposition* reactions are shown in Equations (2.3) and (2.4), respectively.

$$A + B \rightarrow AB \tag{2.3}$$

$$AB \rightarrow A + B \tag{2.4}$$

2.1.1 DNA Replication

All genetic information is encoded by molecules called *nucleic acids.* There are two types of nucleic acids: *deoxyribonucleic acid* or DNA and *ribonucleic acid* or RNA. Both DNA and RNA contain multiple genes that encode and carry genetic information. DNA molecules contain the copy of the cell's genome and can carry thousands of genes to transmit this information. In contrast, RNA molecules are shorter and are used to transport genetic information to the cell machinery carrying only one or a few genes. DNA and RNA are made of smaller subunits called *nucleotides.* There are four different types of nucleotides in each nucleic acid and their arrangement determines the genetic information. Nucleotides corresponding to DNA and RNA are called **A, G, C, T** and **A, G, C, U**, respectively.

Genetic information is passed from parents to children by means of parental DNA *replication.* DNA replication is the process by which a cell's genome is duplicated. Though it might seem simple, replication is a complex process that involves several specialized molecules and proteins. However, DNA replication can be described as two distinct stages: first, the two strands of parental DNA are separated; the second stage consists of making new copies using the two single strands as templates [15].

In the first step of DNA replication, *helicase*, a special protein, unwinds the DNA double helix. Next, *DNA polimerase*, an enzyme that helps in DNA synthesis, binds to one *strand* of DNA and begins to copy in the 3' to 5' direction. The synthesized single-stranded DNA is called the *leading strand* and is used for reforming the DNA double helix. A second DNA polymerase is used to copy the other DNA strand from 5' to 3', which is the direction in which DNA synthesis can only occur. Discontinuous segments of DNA called *okazaki fragments* are synthesized by this second molecule and *DNA ligase*, another enzyme, stitches these fragments together into the *lagging strand* as shown in Figure 2.1.

2.1.2 Central Dogma: On Protein Synthesis

The *central dogma* or doctrine of the triad states that once genetic information has been transformed into a protein, it cannot be converted back into its original state. That is, the transfer of information from nucleic acid to nucleic acid or nucleic acid to protein is possible, but conversion from protein to protein or protein to nucleic acid is impossible [16]. In other words, DNA can be synthesized from itself, RNA can also be synthesized from both DNA and itself, and protein can be synthesized from DNA and RNA. However, protein can never be converted back into DNA or RNA. This process is illustrated in Figure 2.2 and Figure 2.3.

Synthesis of RNA from DNA occurs through a process called *transcription*. This process is akin to DNA replication in that double-stranded DNA is unwound and single strands are used as templates to produce RNA. The main enzyme that directs transcription is *RNA polymerase* or RNAP. To start transcription, RNAP recognizes and binds to a specific site in the beginning sequence of a gene called *promoter*. A promoter sequence is found on one strand of DNA and indicates RNAP where to start transcription and in which direction it should synthesize. Then, RNAP unwinds the double-stranded DNA and starts synthesis of *messenger RNA*, mRNA, in a unidirectional manner. mRNA is also known as *antisense* or *template strand* as it is a single strand complementary to one of the strands of DNA. The other strand is known as the *coding* strand. Transcription terminates when RNAP reaches a region known as the *termination region*. This process is done in two steps: first, the newly formed mRNA is released from RNAP and, second, RNAP itself disengages from DNA.

Transcription factors are proteins that either enhance or inhibit the ability of RNAP to initiate transcription. A transcription factor binds to the *operator site*, a DNA sequence near the promoter, to help RNAP bind to the promoter and activate transcription or block RNAP from attaching to the promoter thus inhibiting the initiation of transcription. Transcription factors that enhance the ability of RNAP to bind to the promoter are called *activators* and those that preclude it are known as *repressors*. Put in other way, an activator “turns on”

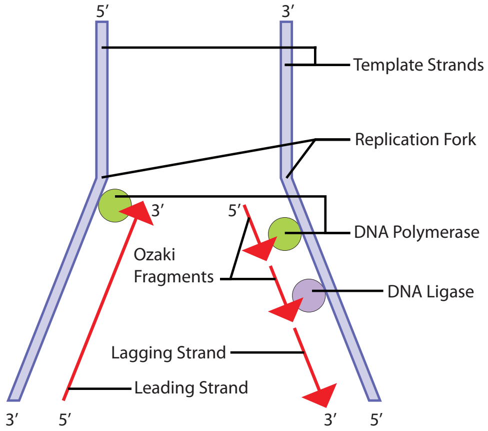


Fig. 2.1: An overview of DNA replication (Courtesy of the National Center for Biotechnology Information).

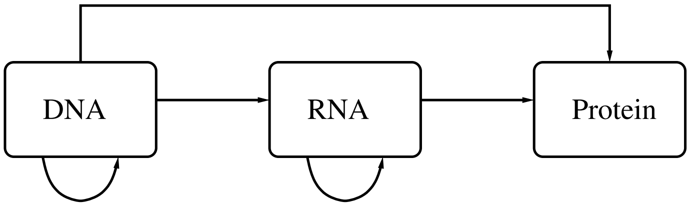


Fig. 2.2: Illustration of protein synthesis flow as stated by the doctrine of the triad or central dogma. The flow of information follows the direction of the arrows.

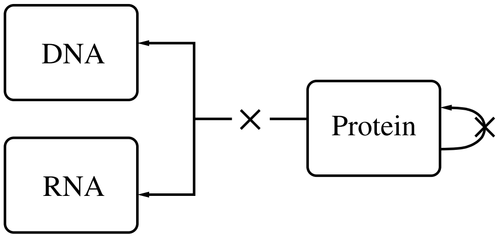


Fig. 2.3: Picture illustrating how proteins cannot be converted back into DNA or RNA as stated by the central dogma. The flow of information follows the direction of the arrows.

transcription and a repressor “turns off” gene expression. When transcription is terminated, protein synthesis starts by the process of *translation.*

Translation involves three important steps: *initiation, elongation*, and *termination.* Each tRNA molecule has two sites called *acceptor site* and *anti-codon site.* The acceptor site binds a particular triplet of nucleotides called *codon* and the anti-codon site binds a sequence of three unpaired nucleotides called *anti-codon.* The codon that signals initiation of translation is ATG which codes for the amino acid *methionine.* Not every protein starts with methionine, though, since this amino acid is oftentimes removed in protein post-processing. Next, elongation begins when a tRNA charged with methionine binds to the translation start codon and the large subunit binds to both mRNA and the small subunit.

After elongation starts, ribosomes shift the first methionine charged tRNA from the A site to the P site. The A site is occupied by a new charged tRNA molecule corresponding to the codons of the mRNA and the two amino acids form a bond. The first tRNA is released and ribosomes shift again so that the P site contains a tRNA with two amino acids. A new charged tRNA binds to the A site and elongation continues on until a stop signal, called *stop codon*, is reached. When the stop codon is found, the ribosome breaks apart into its large and small subunits thus releasing both the new protein as well as the mRNA. This new protein is then ready to undergo *post-translational* modification. The process of transcription and translation is depicted in Figure 2.4.

2.2 Synthetic Biology

Scientists have been modifying biological organisms for many years. In the 1970s, a new engineering discipline was introduced: *genetic engineering*, which permits the modification of an organism’s DNA or genome using *recombinant DNA technology.* In 1972, Paul Berg developed and described a method that allowed the combination of duplex DNA molecules [17]. Berg created the first recombinant DNA molecules when he combined DNA from the monkey virus *SV40* with that of the *lambda phage* virus. Despite the significant advances genetic engineering has introduced in the biomedical sciences, biotechnology, and key understanding of biological organisms, there are still limitations that have yet to be

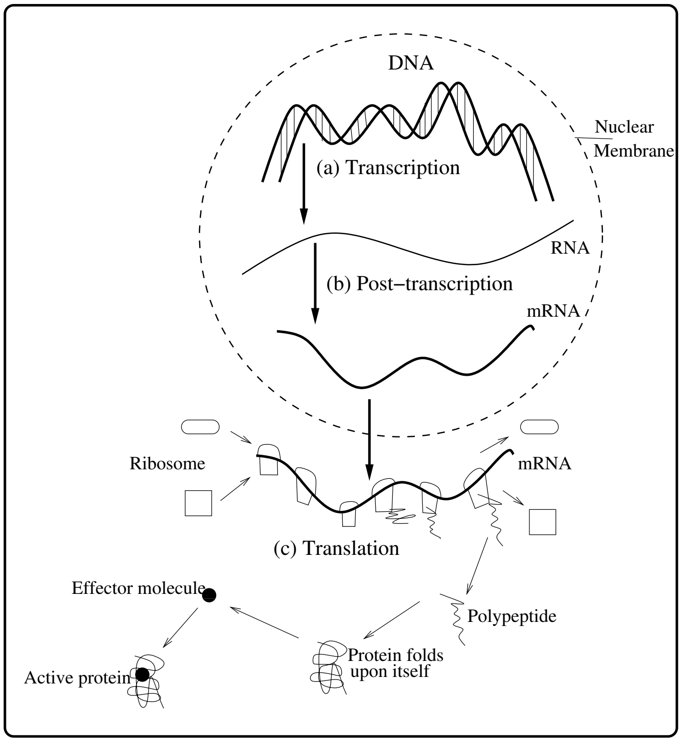


Fig. 2.4: Illustration of protein synthesis depicting the process of transcription and translation along with post-translation processing. Double-stranded DNA is unwound and copied to produce a single strand of RNA through the process of transcription. In post-transcription processing, messenger RNA, or mRNA, is formed and transported out of the nuclear membrane of the cell. Next, translation starts when ribosomes attach to mRNA to begin synthesis of the polypeptide chain encoded in mRNA. This polypeptide chain folds upon itself to form a protein that is active after an effector molecule binds to it.

overcome. Genetic engineering lacks the rigor of disciplines such as electrical engineering. The emergent field of *synthetic biology* takes a step further from simple gene manipulations to the construction of synthetic genomes or *genetic circuits* that can be inserted in organisms to control cell behavior.

According to Drew Endy, professor at Stanford University, synthetic biology aims to make biology easy to engineer. It accomplishes this goal by bringing together the expertise of professionals from biology, chemistry, physics, and engineering [17]. Synthetic biology also seeks to understand life, to introduce new functions in living organisms and make them perform desired tasks, and ultimately, to build life from scratch. For over 30 years genetic engineering has been based on the following three techniques:

- *Recombinant DNA* - construction of artificial DNA through combinations;
- *Polymerase Chain Reaction* (PCR) - copy and amplification of DNA segments;
- *DNA Sequencing* - determining the order of nucleotides of a DNA segment.

As pointed out by Endy [18], synthetic biology extends genetic engineering by adding the following three ideas:

- *Standards* - creation of repositories of parts and devices that can be easily assembled;
- *Decoupling* - division of complex problems into simpler problems to be worked independently (i.e., separating design from construction);
- *Abstraction* - management of complexity by hiding information as depicted in Figure 2.5.

A useful way to explain this nascent field is by drawing an analogy with electrical and computer engineering, as illustrated in Figure 2.6. Historically, electrical engineering emerged from physics as electrical engineers were tinkering with circuits instead of studying relativity or quantum mechanics. In the same way, synthetic biology has splintered off from biology and bioengineering because instead of studying natural organisms, synthetic biologists are engineering new ones. This analogy goes beyond historical relationships, though. In many ways synthetic biologists have based their field on electrical engineering and other fields. Design of new behavior in biological organisms occurs at the top of the hierarchy in Figure 2.6, but the implementation takes place from bottom up. The bottom layer of the hierarchy is comprised of DNA, RNA, proteins, and metabolites such as lipids, carbohydrates, amino acids, and nucleotides [19]. This layer is the counterpart to the physical layer of resistors, capacitors, and transistors in electrical and computer engineering. The next layer up is responsible for the regulation of information through biochemical reactions, equivalent to logic gates in computer systems. Biological devices can then be assembled into modules or complex pathways to function much like integrated circuits (IC) in computers. These pathways can be interconnected and inserted in host cells to program their behavior like

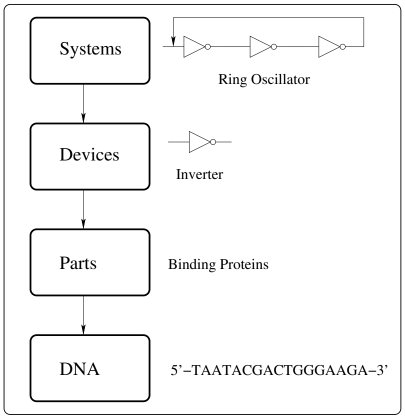


Fig. 2.5: Proposed abstraction hierarchy for the engineering of integrated biological systems. In this figure "DNA" is the genetic material, "Parts" are specialized biological components that perform certain functions, "devices" are combinations of "Parts," and "Systems" are any combinations of "Devices." Each abstraction layer builds from the layer underneath it except for that of "DNA." Every layer hides detailed information of the layer below it to make design easier for the engineer.

microcontrollers in a computer. In consequence, programmed cells can be made to function together as a quorum to perform more sophisticated tasks similar to computer networks.

Synthetic biology promises great benefits for our society by potentially changing living organisms in ways never seen before. However, as is the case with other disciplines, these benefits do not come free of risks. Synthetic manipulation of biological systems can be dangerous for human health and/or the environment [20]. For instance, DNA synthesis has recently helped resurrect the 1918 influenza strain [21, 22], and is also capable of producing the smallpox virus from DNA sequence information available to the public [23]. In addition, the development of this technology can be used by terrorists to fabricate bio-weapons to attack populations. Clearly, safety regulations must be taken by government agencies around the globe to ensure that the use of synthetic biology is confined as much as possible for the development of products beneficial to any form of life of earth.

2.3 Genetic Circuits

DNA segments can be manipulated to bahave in logic ways and build programmed behavior in cellular networks. However, in order to program and rubustly control cell behavior

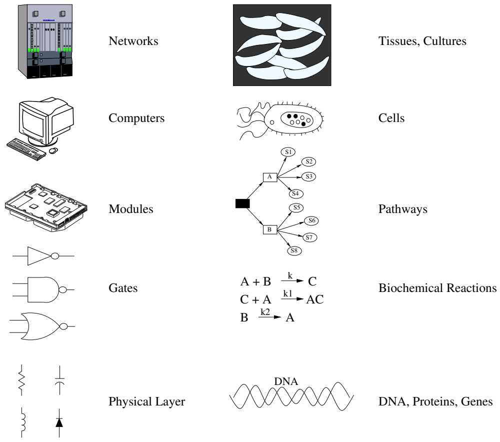


Fig. 2.6: Analogy of hierarchy between synthetic biology and computer engineering. This figure shows a conceptualization of the similarities between synthetic biology and computer engineering. DNA, proteins and metabolites are put side-to-side with resistors, capacitors, transistors, etc. Computing devices such as logic gates are compared to biochemical reactions; modules and pathways, computers and cells, networks and cell populations describe the close relationship these two fields share in common.

it is of utmost importance the creation of a library of well-defined components that serve as the building blocks of more complex systems [24]. Several synthetic gene networks have been designed and embedded in living matter such as toggle switches, logic gates, pattern-forming circuits, oscillators, cellular sensors, and cell-to-cell communication systems [25]. When correctly assembled, certain genetic elements can be configured to implement logic gates and circuits where chemical concentrations of DNA proteins and inducer molecules are the input/output signals as opposed to a stream of ones and zeros. In this section common logic gates are presented to serve as a basic description of simple genetic circuits.

2.3.1 Genetic Inverter

The simplest logic gate is the *inverter*. An example of the biochemical inverter is illustrated in Figure 2.7. Here, the input signal is the TetR protein and the output is the *green fluorescent protein*, GFP, shown schematically in Figure 2.7(a). The presence or absence of TetR determines the two possible output states as shown in Figure 2.7(c). If TetR is present, then GFP production is repressed. On the other hand, if TetR is absent, GFP is transcribed and thus seen at the output of the inverter. Figure 2.7(b) shows the genetic schematic of this device.

When an odd number of inverters are connected in cascade they form an oscillator, as shown in Figure 2.8(a). Oscillators are commonly used to synchronize the behavior of a group of cells. For instance, circadian rhythms vary the concentrations of proteins periodically within a cell [3]. The mechanisms that control the oscillations are unknown to date. However, Elowitz and Leibler constructed the circuit in Figure 2.8(b) which they called the *repressilator*, showing oscillatory fluorescent intensity [26].

2.3.2 Genetic NAND

Another widely used combinational logic gate is the *NAND* gate. This particular gate is known as the *universal gate* because all other gates can be constructed with a set of NANDs. Design and construction of the genetic NAND have been reported by several studies in the literature [24, 27]. The schematic symbol used to represent the NAND gate is shown in Figure 2.9(a). The two-input NAND gate consists of two separate inverters with different inputs but with the same output, as illustrated in Figure 2.9(c). The output of the NAND gate is always HIGH unless both input signals are present. This behavior is described by the truth table in Figure 2.9(b).

2.3.3 Genetic Toggle Switch

A *toggle switch* is a circuit that exhibits two possible states. Also referred to as *bistable switch*, the state of the genetic toggle switch can be either *high* or *low*. Gardner *et al.* designed and constructed a synthetic, bistable gene-regulatory network in *Escherichia coli*,

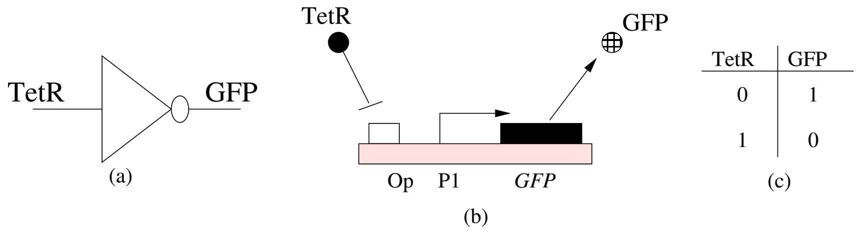


Fig. 2.7: Description of the genetic inverter: (a) schematic symbol, (b) genetic implementation, and (c) truth table.

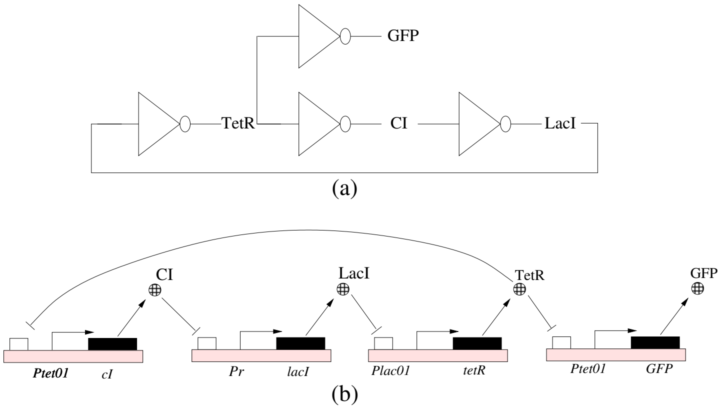
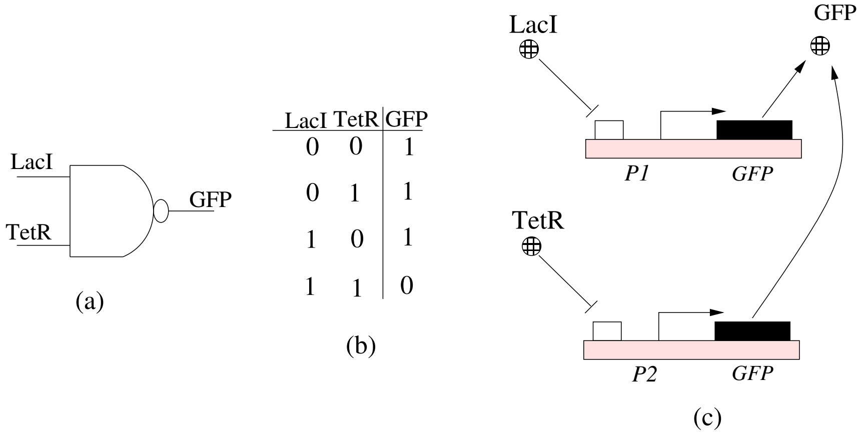


Fig. 2.8: Genetic ring oscillator. This oscillator is formed by cascading an odd number of inverters: (a) Logic diagram, (b) Corresponding genetic implementation.



LacI	TetR	GFP
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 2.9: Genetic NAND gate: (a) Schematic symbol, (b) Truth table, (c) Genetic implementation.

giving it the toggling behavior using two repressible promoters assembled in a mutually inhibitory network [10], as shown in Figure 2.10.

2.4 Modeling Genetic Circuits

The purpose of research in systems/synthetic biology is basically to provide a wider understanding about the behavior and workings of both natural and synthetic biological systems [28]. Thorough knowledge of biological systems would translate into their successful exploitation for agricultural, medical, energy, commercial, and other purposes. State of the art technology in the biological sciences has allowed the construction of virtually any DNA sequence. Nevertheless, prediction of the behavior of synthetic biological systems is not easy to accomplish. Modeling the behavior of genetic circuits before they are synthesized is an essential part synthetic biology. Similar to silicon chip fabrication, where modeling tools are used to help guide design, genetic circuit fabrication can benefit from modeling tools that can predict the dynamic characteristics of the behavior of proposed circuits [24]. The simulation tools can be used to design and verify functions of various circuit network configurations, thus minimizing production time and effort as well as well as cost.

Reproduction of some properties displayed by a biological system is achieved by the creation of mathematical and computational models that are then used to try to predict their behavior. As outlined by Klipp *et al.* [28], the main reasons for modeling a biological system include:

- Testing accuracy of the model. Does this model closely reflect known experimental facts?
- Analysis of model to understand the parts of the system that contribute most to the properties of interest.
- Hypothesis generation and testing to allow rapid analysis of the effects of manipulating experimental conditions in the model without performing complex and expensive experiments.

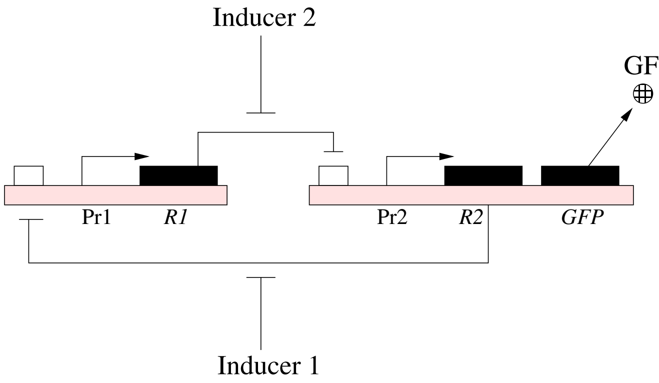


Fig. 2.10: Genetic toggle switch design. Repressor R1 represses transcription of promoter Pr2 and is induced by Inducer 1. Repressor R2 inhibits transcription of promoter Pr1 and is induced by Inducer 2. Green fluorescent protein (GFP) is used as the reporter protein.

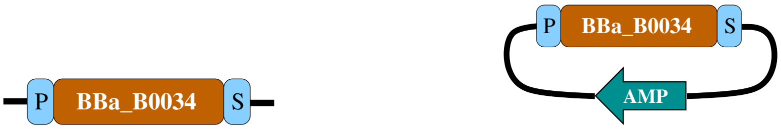
- Testing and verifying what changes in the model would improve the consistency of its behavior with the experimental observations.

In general, models are abstractions of reality represented using diagrams, laws, graphs, plots, mathematical relationships, chemical formulae, and so on, that try to describe and understand some external physical phenomena. In systems biology, models try to describe the relationship between metabolites or signaling molecules that interact through chemical reactions. These models are often composed of chemical reaction networks including mathematical equations describing the local behavior and the values of all parameters [29]. The classical approach of modeling biochemical systems is by means of ordinary differential equations (ODE). In order to use ODEs one must know, first, the set of chemical reactions that govern the system along with its effector molecules; second, the kinetic rate equations that relate the rate of each reaction to the concentrations of its substrates, and finally, the parameterization of the model providing values for the parameters [29]. With this knowledge, software tools can simulate and predict the time evolution of the system until it reaches a steady state. This is accomplished by solving a system of ordinary differential equations to which the system is recast.

Even though ODEs are well documented and easy to solve with several methods, there are issues with this approach. First, ODE models assume that molecules in cell compartments are well stirred and that the concentrations (number) of molecules is sufficiently large to ignore fluctuations. ODE models assume that the system varies continuously and deterministically, which is not true in the real scenario. Stochastic simulations are amenable when the reactions in the system are considered to fire in a random manner as well a the existence of a discrete number of molecules. Ordinary differential equation (ODE) models and stochastic simulation methods will be discussed in depth in the following chapter.

2.5 BioBricks: Building Genetic Circuits

Development of standard components and processes has allowed mature engineering disciplines to accomplish major advances. Synthetic biology can also benefit from the development and implementation of standard components and practices. For this reason, the BioBricks Foundation has been established to create a registry of standard biological parts [30, 31]. These parts, commonly called *BioBricks*, include *terminators, ribosome binding sites, protein coding regions (i.e., genes), reporter genes, signaling parts, regulatory sequences, gates*, etc. A BioBrick illustration is shown in Figure 2.11(a). In order to be functional, a BioBrick is inserted into a plamid DNA along with an antibiotic, as shown in Figure 2.11(b). BioBricks can be assembled to form much more complex devices. The assembly process is depicted in Figure 2.12. Each BioBrick has four domains: E and X located upstream (before) the BioBrick, and S and P positioned after the BioBrick. The first BioBrick, labeled *upstream part* in Figure 2.12, is cut with restriction enzymes that cut domains E and S while the second BioBrick is cut at the E and X domains. The results are then mixed and ligated (glued) into the destination plasmid, producing the final combined BioBrick. More information on BioBrick standards, protocols, and assembly can be found in parts registry website (http://www.partsregistry.org).



(a) BioBrick part.

(b) BioBrick inserted in plasmid DNA along with an antibiotic.

Fig. 2.11: BioBrick examples.

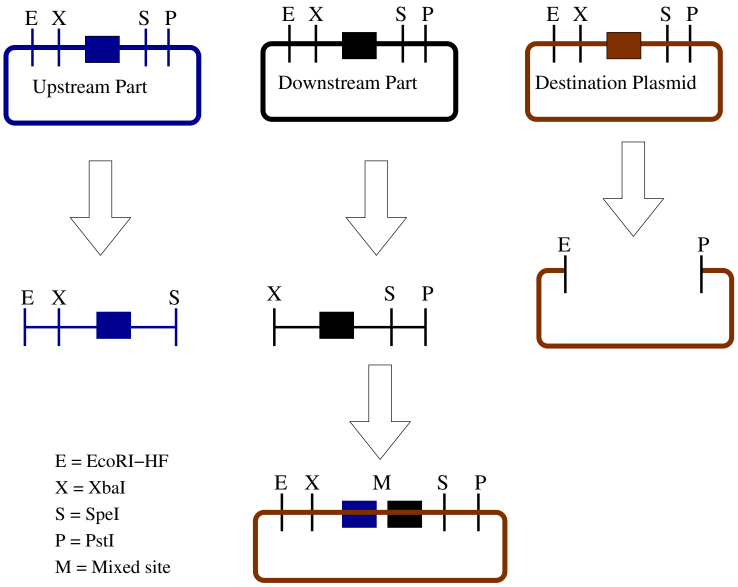


Fig. 2.12: Depiction of BioBrick assembly process. Courtesy of the BioBrick Foundation http://www.partsregistry.org.

Chapter 3

Math Methods and Algorithms for Simulation of Genetic Circuits

After a model of a biological system has been constructed, it is often analyzed by means of *in silico* (computer) simulations. The aim of computer aided simulations is to make predictions of system behavior that has not yet been observed in a laboratory setting. *In silico* simulations offer the benefit of unlimited controlability and observability allowing the designer to gain deep insight about the biological system under consideration which would be otherwise more difficult.

Different mathemcatical methods and computational algorithms have been developed to analyze and simulate biological system models *in silico*. This chapter provides a brief overview of the most relevant methods and algorithms utilized in genetic circuits modeling. This chapter is organized as follows: Section 3.1 describes the *classical chemical kinetics* ODE model and Section 3.2 presents several stochastics methods used to simulate biological models.

3.1 Ordinary Differential Equation (ODE) Model

In 1864, Waage and Guldberg introduced the *law of mass action*, which was later translated by Abrash in 1986 [32]. As described in Section 2.1 on page 6, this law states that the reaction rate is proportional to the concentration of reactants. Using the law of mass action, a chemical reaction network can be translated into a system of ordinary differential equations (ODE). These differential equations are commonly known as *reaction rate equations.* Traditional *classical chemical kinetics* (CCK) makes use of ODEs to represent system dynamics. Since systems of ODEs are generally difficult to solve analytically, numerical simulations are more than often used to determine the behavior of the chemical model.

A CCK model follows the concentration of every species in the system. There are some assumptions that must be satisfied in order for the CCK model to be valid. First, the CCK model assumes that all reactions accur in a *well-stirred* volume. That is, the molecules are evenly distributed. This assumption implies that *spatial effects* are neglected. Finally, the CCK model assumes that reactions occur continuously and deterministically, meaning that the number of molecules in the cell must be very large. These assumptions along with the law of mass action are used to derive an ODE model that can describe the dynamics of certain biochemical systems.

A chemical reaction system is composed of n chemical species $\{S_1, ..., S_n\}$ and m chemical reaction channels $\{R_1, ..., R_m\}$. Following the notation used by Myers [3], each reaction can be written as

$$v^r_{1j}S_1 + ... + v^r_{nj}S_n \underset{k_r}{\overset{k_f}{\rightleftarrows}} v^p_{1j}S_1 + ... + v^p_{nj}S_n, \tag{3.1}$$

where v^r_{ij} is the reactant stoichiometry coefficient for species S_i in reaction R_j and v^p_{ij} is the product stoichiometry coefficient for species S_i in reaction R_j. If species S_i does not participate in reaction R_j then the value of v^r_{ij} or v^p_{ij} is zero. k_f is the forward rate constant while k_r is the reverse rate constant. If reaction R_j is irreversible, then k_r is 0.

The rate of a reaction R_j can be written mathematically as

$$V_j = k_f \prod_{i=1}^{n} [S_i]^{v^r_{ij}} - k_r \prod_{i=1}^{n} [S_i]^{v^p_{ij}}, \tag{3.2}$$

where $[S_i]$ is the concentration of species S_i. This equation is the mathematical representation of the law of mass action expressing that the rate of an irreversible ($k_r = 0$) chemical reaction is proportional to the product of the concentrations of the reactant molecules. If the reaction is reversible, then the rate is also reduced by a value proportional to the product of the concentrations of the product molecules. If the net change in species S_i is denoted as $v_{ij} = v^p_{ij} - v^r_{ij}$, then Equation (3.2) can be used to construct an ODE model as follows

$$\frac{d\,[S_i]}{dt} = \sum_{j=1}^{m} v_{ij}V_j, \quad 1 \leq i \leq n. \tag{3.3}$$

The ODE model in Equation (3.3) creates a differential equation for each species in the reaction network as the sum of the rates of change of the species due to each reaction affecting the species. Michaelis-Menten enzymatic reaction system [33] is a simple yet well-known model that can be used as a working example. Figure 3.1(a) shows the chemical reaction network for the Machaelis-Menten system. The corresponding reaction rate equations are shown in Figure 3.1(b) and Figure 3.1(c) outlines the corresponding system of differential equations (ODEs). A set of ODEs like that shown in Figure 3.1(c) is very hard if not impossible to solve analytically. Several numerical methods can be used to approximate the time evolution of such a sytem. The reader is advised to study methods such as the *Euler's method, backward Euler method, Runge-Kutta*, among others found in most college calculus textbooks. Figure 3.2 shows the simulation results correspoding for the Michaelis-Menten reaction network.

$$\begin{aligned} E+S &\overset{k_1}{\rightarrow} ES \\ ES &\overset{k_{-1}}{\rightarrow} E+S \\ ES &\overset{k_2}{\rightarrow} E+P \end{aligned} \qquad \begin{aligned} V_1 &= k_1\,[E]\,[S] \\ V_2 &= k_{-1}\,[ES] \\ V_3 &= k_2\,[ES] \end{aligned}$$

(a) (b)

$$\begin{aligned} \frac{d[E]}{dt} &= -k_1[E][S] + (k_{-1}+k_2)[ES] \\ \frac{d[S]}{dt} &= -k_1[E][S] + k_{-1}[ES] \\ \frac{d[ES]}{dt} &= k_1[E][S] - (k_{-1}+k_2)[ES] \\ \frac{d[P]}{dt} &= k_2[ES] \end{aligned}$$

(c)

Fig. 3.1: Michaelis-Menten enzymatic reaction network ODE model. (a) Chemical reaction network, (b) Reaction rates, and (c) ODE model for the Michaelis-Menten enzymatic reaction system.

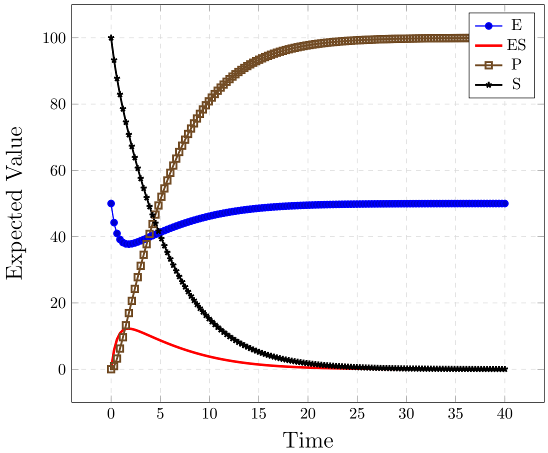


Fig. 3.2: Michaelis-Menten ODE simulation results. All network parameters to produce this results are outlined in Appendix A.1. This simulation was done using iBioSim, a CAD tool under development by Dr. Chris Myers' group at the University of Utah.

3.2 Stochastic Models

As was described in Section 3.1, a chemical reaction network can be written as a set of first-order ordinary differential equations (ODEs). An ODE model is valid only under the assumptions that molecule concentrations vary continuously and deterministically. However, biochemical systems do not satisfy neither of these assumptions. In chemical systems formed by living cells, there is a small number of molecules for each species. Thus, the system can show a dynamical behavior that is discrete and stochastic as opposed to continuous and deterministic [7, 34–36]. A chemical reaction is typically fired after two or more molecules collide. Hence, anticipation of a chemical reaction is almost impossible to achieve unless one is able to precisely track the position and velocity of the molecules. In consequence, stochastic modeling methods are amenable and preferred to describe the dynamics of biochemical systems.

The foundations of stochastic modeling start with the development of a *stochastic chemical kinetic* (SCK) model, which has lead to the creation of the *chemical master equation* (CME). Unfortunately, the CME cannot be solved either analytically nor numerically for most systems. However, *Monte Carlo* simulation methods, such as the *Stochastic Simulation Algorithm* (SSA), can generate numerical realizations of the chemical master equation. Building on and improving SSA, several other methods have been created such as *tau leap-*

ing, Gibson-Bruck's next reaction method, among others [6,37]. This section will address the foundations for stochastic modeling by briefly explaining the SCK, CME, and SSA models.

A SCK model is described as follows. Recall that a biochemical reaction network is composed of n chemical species $\{S_1, ..., S_n\}$ and m chemical reactions $\{R_1, ..., R_m\}$. The system is assumed to be contained in a constant volume Ω, is *well-stirred*, and in *thermal equilibrium* (constant temperature). Let $X_i(t)$ denote the number of molecules of species S_i at time t. It follows that the state of the system at time t can be expressed in vector form as $\mathbf{X}(t) = (X_1(t), ..., X_n(t))$. It is desirable to study the time evolution of the system, given that the system was in some initial state $\mathbf{X}(t_0) = \mathbf{x_0}$.

The reaction channel is described mathematically by a two-dimensional array known as the *stoichiometric matrix.* Each row of this matrix is formed by a *state-change vector* $\mathbf{V}_j = (v_{ij}, ..., v_{nj})$, where v_{ij} is the molecular change of species S_i due to reaction R_j. If a reaction R_j occurs, then the sytem is updated to state $\mathbf{x} + \mathbf{v}_j$. Each reaction channel R_j is considered to be elemental, meaning that chemical reactions are eiher unimolecular or bimolecular, taking v_{ij} values of $0, \pm1$, and ±2. Elemental reactions are considered to happen essentially instantaneously. In addition to the stoichiometric matrix, each reaction channel has associated with it a *propensity function* a_j. The propensity function is defined such that $a_j(\mathbf{x})$ is the probability that reaction R_j occurs somewhere in Ω in the next infinitesimal time interval $[t, t + dt]$, given $\mathbf{X}(t) = \mathbf{x}$. In practice, this probability is found by multiplying the number of possible reactant molecules by c_j, a *specific probability rate* related to the reaction rate constant k. This probabilistic definition of the propensity function has been justified in physical theory by Gillespie [6].

For a unimolecular reaction $S_i \rightarrow P$, there exists some constant c_j such that the propensity $a_j(\mathbf{x}) = c_j x_i$, where x_i is the number of molecules of species S_i. The computation of c_j for this type of reaction requires consideration of quantum mechanics. If R_j is a bimolecular reaction $S_1 + S_2 \rightarrow P$, there exists a different constant c_j and the propensity function is given by the multiplication of c_j and the possible number of combinations of S_1 and S_2 molecules that can react $x_1 x_2$, $a_j(\mathbf{x}) = c_j x_1 x_2$. If $x_1 = x_2$ the possible number of combinations of is given by $\frac{1}{2}x_1(x_1-1)$ and $a_j(\mathbf{x}) = c_j\frac{1}{2}x_1(x_1-1)$. It has been shown that for monomolecular reactions c_j is numerically equal to the reaction rate constant k_j of conventional chemical kinetics, while for bimolecular reactions c_j is equal to k_j/Ω for different reactant species and $2k_j/\Omega$ if the species are the same [38].

Using the probabilistic theory above, the probability $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ that $\mathbf{X}(t)$ will be in state $\mathbf{x}$ at time t given the initial state $\mathbf{X}(t_0) = \mathbf{x}_0$ can be formulated as

$$\begin{aligned} P(\mathbf{x}, t+dt|\mathbf{x}_0, t_0) =& P(\mathbf{x}, t|\mathbf{x}_0, t_0)\left[1-\sum_{j=1}^{M} a_j(\mathbf{x})dt\right] \\ &+ \sum_{j=1}^{M} P(\mathbf{x}-\mathbf{v}_j, t|\mathbf{x}_0, t_0) \times a_j(\mathbf{x}-\mathbf{v}_j)dt, \end{aligned} \tag{3.4}$$

where the first term on the right is the probability that the system is already in state $\mathbf{x}$ at time t and no reaction occurs in the infinitesimal time interval $[t, t+dt]$; the second term on the right is the probability that the system is $\mathbf{v_j}$ away from state $\mathbf{x}$ and reaction R_j occurs in the next time interval $[t, t+dt]$. dt must be chosen to be sufficiently small such that only one reaction can occur in the time period $[t, t+dt]$. By subtracting $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ from both sides of Equation (3.4), dividing it by dt, and taking the limit $dt \to 0$ [39] the *Chemical Master Equation* is written as

$$\frac{\partial P(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\partial t} = \sum_{j=1}^{M} [a_j(\mathbf{x}-\mathbf{v}_j)P(\mathbf{x}-\mathbf{v}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x})P(\mathbf{x}, t|\mathbf{x}_0, t_0)]. \tag{3.5}$$

In theory, this differential equation determines completely and exactly the function $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$. However, it cannot be solved analytically or numerically except for very few simple systems because this equation represents a set of nearly as many coupled differential equations as there are molecule combinations that exist in the system [40]. Since the CME is not of much use to compute the probability density function $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$ of $\mathbf{X}(t)$, another more feasible computational approach is needed. That leads us to the introduction of Gillespie's *Stochastic Simulation Algorithm* (SSA) in the following section.

3.2.1 Gillespie's Stochastic Simulation Algorithm (SSA)

Since the Chemical Master Equation (CME) cannot be solved either analytically or numerically for most interesting biological systems, another formulation is required to be able to simulate biochemical reaction networks *in silico.* One approach that has proven to be effective is the simulation of trajectories or samples of $\mathbf{X}(t)$ versus time. Note that this is not the same as numerically solving the CME, but just taking samples of that random variable. Nevertheless, the same effect can be achieved by averaging the trajectories of many realizations. In order to generate simulated trajectories of $\mathbf{X}(t)$, a new function $p(\tau, j|\mathbf{x}, t)$ is defined such that $p(\tau, j|\mathbf{x}, t)dt$ is the probability that the next reaction to occur in the next infinitesimal time interval $[t+\tau, t+\tau+dt]$ is R_j assuming the current state is $\mathbf{X}(t) = \mathbf{x}$ [38]. This newly formulated function is a joint probability density function for the two random variables "time to the next reaction," τ, and the "reaction index," j.

An analytical expression can be derived for $p(\tau, j|\mathbf{x}, t)$ by defining the function $P_0(\tau, \mathbf{x}, t)$ as the probability that no reaction occurs in the time interval $[t, t+\tau]$ given $\mathbf{X}(t) = \mathbf{x}$. Using the laws of probabilities a new relationship can be obtained

$$p(\tau, j|\mathbf{x}, t)dt = P_0(\tau, \mathbf{x}, t) \times a_j(\mathbf{x})d\tau. \tag{3.6}$$

Another implication from the laws of probability is

$$P_0(\tau + d\tau|\mathbf{x}, t) = P_0(\tau|\mathbf{x}, t) \times [1 - \sum_{j'=1}^{M} a_j(\mathbf{x})dt]. \tag{3.7}$$

Using Equation (3.7) a differential equation can be written as

$$\frac{d\mathcal{P}_0(\tau, \mathbf{x}, t)}{d\tau} = -a_0(\mathbf{x})\mathcal{P}_0(\tau|\mathbf{x}, t), \tag{3.8}$$

where $a_0(\mathbf{x}) = \sum_{j=1}^{M} a_j(\mathbf{x})$. The solution to Equation (3.8) with initial condition $\mathcal{P}_0(\tau = 0|\mathbf{x}, t) = 1$ is

$$\mathcal{P}_0(\tau|\mathbf{x}, t) = exp[-a_0(\mathbf{x})\tau]. \tag{3.9}$$

Combining Equations (3.9) and (3.6) and canceling $d\tau$ results in

$$\begin{aligned} p(\tau, j|\mathbf{x}, t) =& exp[-a_0(\mathbf{x})\tau] \times a_j(\mathbf{x}) \\ =& a_0(\mathbf{x}) exp[-a_0(\mathbf{x})\tau] \times \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})}. \end{aligned} \tag{3.10}$$

Equation (3.10) is the root of the *stochastic simulation algorithm* (SSA). Using Monte Carlo simulations, random samples can be generated for the joint probability function in Equation (3.10) by drawing two random numbers $r1$ and $r2$ from the uniform distribution and selecting τ and j according to the following equations

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r1}\right), \tag{3.11}$$

$$j = \text{smallest integer satisfying} \sum_{j'}^{j} a_{j'}(\mathbf{x}) > r2 a_0(\mathbf{x}). \tag{3.12}$$

With the above theory, the SSA algorithm is formulated as shown in Figure 3.3. The SSA and the CME are equivalent to each other; even though the CME is impractical and intractable, the SSA is easy to implement. Trajectories produced from SSA simulations are stochastic versions of results obtained using ODE methods. When SSA trajectories are close enough to ODE results, then one can conclude that micro-scale fluctuations can be ignored. However, when SSA trajectories deviate a lot from the ODE trajectory, then it follows that micro-scale fluctuations cannot be ignored [40]. The problem with SSA is that it is very computationally expensive making simulations run slow. The computational cost of SSA is found in the computation of Equation (3.11). If the population of one or more reactant species is very large, as it often is, $a_0(\mathbf{x})$ is very large, thus making the simulation time step τ very small. Variations on SSA have been devised to make it more computationally efficient [7, 8, 37].

1. Initialize system state $\mathbf{x} = \mathbf{x}_0$ and time $t = t_0$.
2. Compute the propensity functions $a_j(\mathbf{x})$ at state $\mathbf{x}$, and their sum $a_0(\mathbf{x})$:

$$a_j(\mathbf{x}) = \begin{cases} c_j x_1 x_2 & x_1 \neq x_2 \text{ (Bimolecular reactions)} \\ c_j \frac{1}{2} x(x-1) & x_1 = x_2 \text{ (Monomolecular reactions)} \end{cases}$$

$$a_0(\mathbf{x}) = \sum_{j=1}^{m} a_j(\mathbf{x})$$

3. Draw two uniform random numbers $r1$ and $r2$.
4. Calculate the time to the next reaction τ according to Equation (3.11).
5. Determing the next reaction, R_j using Equation (3.12).
6. Update system state after reaction R_j: $t = t + \tau$ and $\mathbf{x} = \mathbf{x} + \mathbf{v}_j$.
7. If $t < T_{sim}$, where T_{sim} is the simulation time, record state $(\mathbf{x}, t)$, and go back to step 2.
8. End simulation.

Fig. 3.3: Gillespie's Stochastic Simulation Algorithm (SSA).

3.2.2 iSSA: An Incremental Stochastic Algorithm

ODE methods have been shown to have some inadequacies when applied to highly stochastic systems [7]. An intuitive approach to smooth out stochastic simulation results is by taking the average of multiple independent SSA runs. Yet intuitive, this approach is wrong. The designer wants to see a smooth path showing the "typical" behavior of the system, but in many cases, this averaged result is erroneous because stochastic trajectories do not align closely in time in independent simulations. To address this problem the iSSA algorithm has been introduced [11]. iSSA performs independent SSA (or one of its variants) runs in small time increments. At the end of each time interval, statistics are computed and used to constrain the initial condition for the next interval. iSSA uses different techniques to perform statistics. The most common ones presented by Winstead *et al.* [11] are the *marginal probability density evolution* (MPDE) and *mean path* (MP). The results from iSSA simulation correspond to real stochastic simulation runs while showing the designer functional details of the system. Our main focus in this thesis is MPDE which is described in depth in Chapter 4.

Marginal Probability Density Evolution (MPDE) is a method that tracks the statistical evolution of every species in the reaction system. The aim of MPDE is to generate a marginal probability distribution for each species as opposed to a scalar value obtain from traditional SSA simulations or joint statistics from the CME. MPDE is intended to provide statistical information in a way that is intuitively useful for the design of genetic circuits. As part of iSSA, MPDE considers the system's evolution in small increments of time. In other words, the simulation time frame is partitioned into small time-slices. During each slice MPDE approximates all species as a set of independent Gaussian-distributed random variables. Molecule counts are randomly generated at the beginning of each time-increment using each specie's marginal Gaussian probability distribution. After the simulation is terminated, statistics are calculated in the form of mean and variance. MPDE is able to follow the statistical envelope as the system evolves over time, thus providing the designer with a measure of robustness and stability. More details about MPDE is presented in Chapter 4.

Chapter 4

Conservation Constraints Analysis

By definition, the *marginal probability density evolution* (MPDE) algorithm is a method that computes and tracks a marginal probability distribution for each species in a reaction network as opposed to single SSA runs. MPDE is aimed at providing intuitive statistical information to potential bio-designers that need to know the expected behavior of highly stochastic systems. In addition, MPDE provides the designer information about robustness and stability of the system. MPDE is one of the internal methods used by iSSA [11] to compute different statistics of stochastic simulations. In a nutshell, MPDE divides the simulation time in small time increments; during each increment, $MaxRuns$ SSA (or any of its flavors) runs are simulated. At the beginning of each time increment, MPDE approximates all species as a set of independently distributed Gaussian random variables and molecule counts are randomly generated from this distribution. Once the SSA runs are terminated within the current time increment, statistics are computed in the form of mean and variance, a new starting state is generated from the Gaussian distribution, time is advanced to the next time increment, and the process is repeated over until the maximum simulation time *timeLimit* is reached. The steps of this algorithm are shown in Figure 4.1 along with a depiction in Figure 4.2. Doing so, MPDE is able to follow a statistical envelope as the simulation time evolves, while supplying the designer with a measure of confidence and stability.

Figure 4.3(a) shows the results of simulating a toggle switch with SSA. A toggle switch is a bistable system which is characterized by having two distinct states that are commonly referred to as *ON/OFF* or *HIGH/LOW*. The model for this switch was introduced by Wilhelm [41]. These results show how the method of averaging in Figure 4.3(b)is not able to produce the expected behavior of a system that exhibits bi-stability. On the other hand, the true behavior of the switch is captured when simulated with MPDE, as observed in the top curve (labeled MPDE) of Figure 4.3(b). Moreover, MPDE provides the designer information about what is the most likely state this switch can be at steady state. In its original formulation MPDE can only be applied to a narrow array of reaction network models. In order to yield proper results, MPDE relies on the assumption that species are conditionally independently given the rest of the system and that they are distributed as Gaussian random variables. Many of the important models found both in nature and in the lab have tightly correlated species which prevent MPDE from producing the expected results. These correlations will be thoroughly explained in Section 4.2.

Moreover, abstraction methods used to reduce the number of effective reactions and improve computational complexity impose strong correlations among groups of species. These correlations arise from conservation laws that constrain the state of one or more species to vary in terms of other species in the system. In consequence, when applied to large and abstracted systems, MPDE may yield distorted results by violating conservation constraints when the system is approximated with an LGN. In addition to linear conservation constraints there might appear other subtle types of correlations even after correcting errors due to conservation laws. These correlations can be identified and corrected by means of a "correlation matrix" and *principal component analysis* (PCA), respectively. When applied to a data set consisting of interrelated variables, PCA aims to reduce the dimensionality of the data while keeping most of the variation in the data set. This reduction in dimension is achieved by transforming the original data set into a new set of variables, principal components, which are uncorrelated and ordered such that the first few retain most of the variation present in the original variables [42]. Although this work does not treat species correlations other than linear conservation relationships, Section 4.3 describes how PCA can be integrated with MPDE to resolve more subtle relationships among the molecular species.

By resolving linear conservation constraints and using PCA as a run-time verification tool, limitations in MPDE can be greatly reduced, allowing the algorithm to be applied to a wider array of interesting models [43]. The following sections explains the theory behind linear conservation constraints analysis and how it is used with MPDE to keep linear relationships intact during simulation.

```
1. Initialize $t' = 0$, $\boldsymbol{\mu} = \mathbf{0}$, $\boldsymbol{\sigma}^2 = \mathbf{0}$, and $\mathbf{x}' = \mathbf{x}_0$.
2. Initialize $limit = t' + increment$, $run = 1$.
3. Initialize $t = t'$, and $\mathbf{x} = \mathbf{x}' + \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.
4. Execute steps $1 - 7$ of Gillespie SSA algorithm depicted in Figure 3.3 on Page 30.
5. If $run < MaxRuns$, record $\mathbf{x}(limit)$, increment $runs = runs + 1$, and go back to
   step 4.
6. If $t < T_{sim}$, set $t' = limit$, $\mathbf{x}' = \mathbf{x}(limit)$.
7. Set $\boldsymbol{\mu} = mean(\mathbf{x})$, $\boldsymbol{\sigma} = var(\mathbf{x})$.
8. Record $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and go back to step 2.
```

Fig. 4.1: Marginal Probability Density Evolution (MPDE) algorithm.

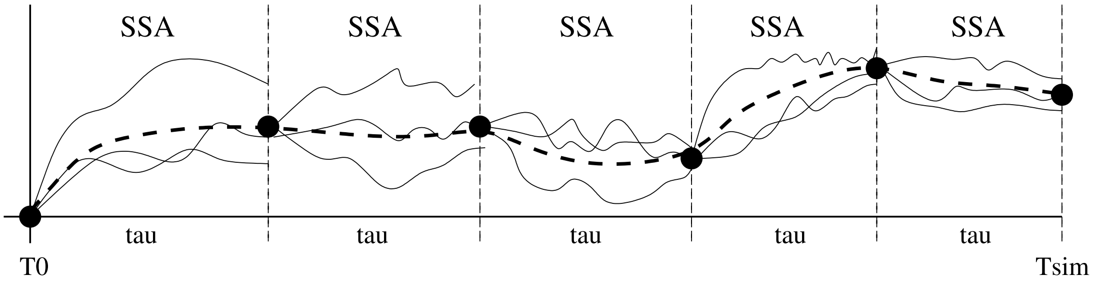
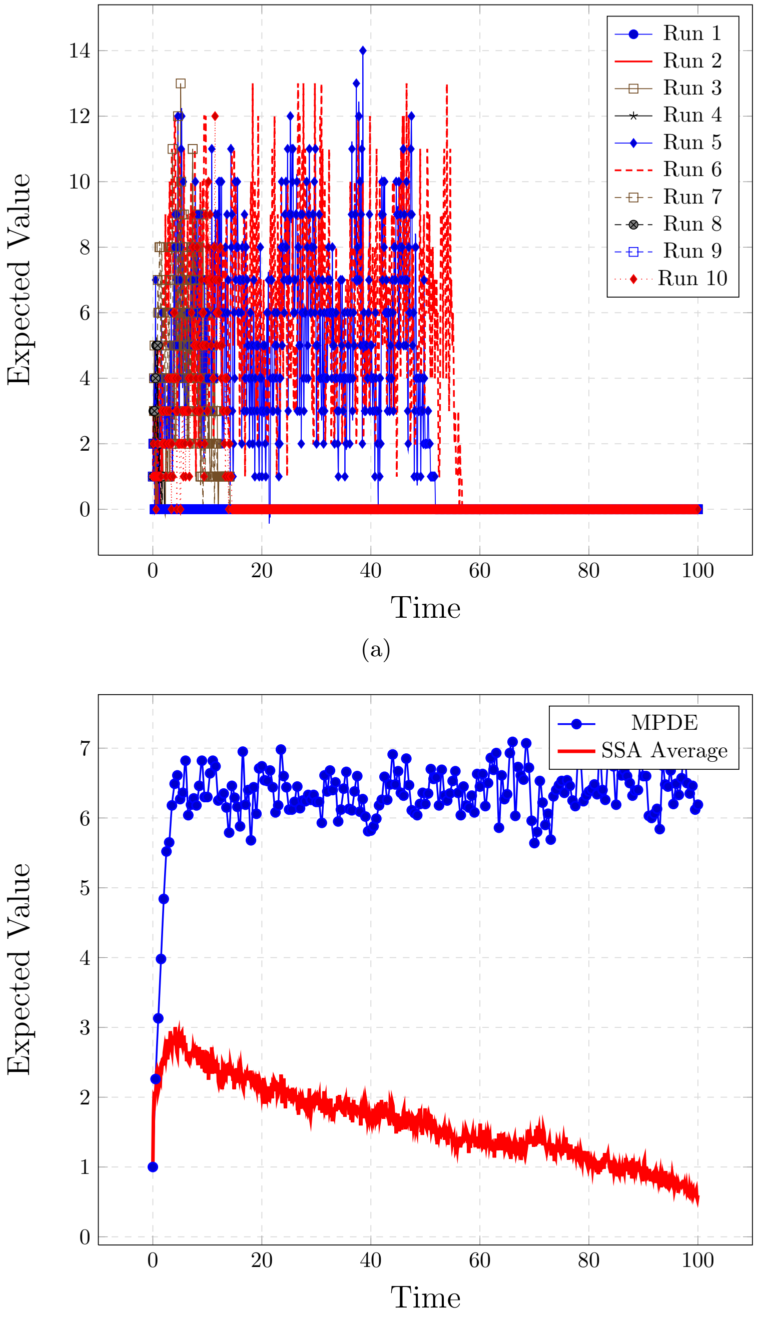


Fig. 4.2: Graphical depiction of the MPDE algorithm steps. Initially, a starting system state $\mathbf{x}_0$ is chosen. Simulation time $Tsim$ is divided into time increments inside which N SSA runs are executed. Statistics are computed at the end of each time increment and used to generate a new state used as the initial state for the next time increment. This process is repeated until the maximum simulation time in reached.



(a)

(b)

Fig. 4.3: Results of simulations of the toggle switch: (a) Ten independent SSA runs of the bistable (toggle) switch, (b) Simulation with MPDE using 100 runs and $\tau = 0.5$ and average of 200 independent SSA runs. The parameters and network description for this model is found in *http://www.ebi.ac.uk/biomodels-main/publmodels* under ID BIOMD0000000233. This simulation data was obtained using iBioSim.

4.1 Linear Conservation Constraints

In this section we show how conserved species can be identified by performing some transformations on the stoichiometric matrix. Using this algorithm, the conservation constraint failure of MPDE can be resolved, thus making MPDE a more robust and efficient method for simulating synthetic genetic circuits.

4.1.1 The Stoichiometry Matrix

The stoichiometric matrix embodies the network topology of any biochemical network [12]. Stoichiometric analysis is not a new study in the field of Systems Biology. Several researchers, like Schuster *et al.* [44], have published pioneering work in the literature since the early 1960s. Any biochemical reaction network can be represented mathematically using the stoichiometry matrix. If a given reaction network is composed of m species and n reactions, then its stoichiometric matrix is a matrix of dimensions mxn. Each row corresponds to a species and each column represents a chemical reaction. In this work $\mathbf{N}$ is used to denote the stoichiometric matrix.

A stoichiometric matrix is shown below. Entry a_{ij}, called stoichiometric coefficient, indicates whether species S_i is affected by reaction R_j or not. The sign of a_{ij} indicates whether it is a reactant or a product. The magnitude reveals the amount of substance that is lost or gained in that particular reaction. This matrix is time-invariant and is solely concerned with the molecular amounts transfered between species according to the chemical reactions that govern the biochemical network [12].

$$\mathbf{N}_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

Consider the biochemical network shown in Figure 4.4. In this system, the reaction network is given by chemical reactions (4.1) and (4.2). In both reactions, the stoichiometric coefficients are equal to 1. In reaction (4.1), S_2 is the reactant species and S_1 is the product.

The stoichiometric matrix for this simple model can be formed as follows: let species S_1 correspond to row 1, species S_2 to row 2, reaction R_1 in column 1, and reaction R_2 in column 2. The stoichiometric matrix is given by

$$\mathbf{N} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Notice that reactants have negative signs and products are accompanied by positive signs. At this point, one might ask how can conserved moieties be extracted from the stoichiometric matrix? Section 4.1.2 explains how this relationships can be spotted using the stoichiometric matrix and matrix algebra.

$$S1 \leftarrow S2 \tag{4.1}$$

$$S1 \rightarrow S2 \tag{4.2}$$

4.1.2 Identifying Conservation Laws

Conserved cycles in a chemical reaction network appear as linear dependencies in the row dimensions of the stoichiometric matrix [12, 45]. In systems where conservation constraints exist, the sum of the conserved species must be constant. For instance, the rate of appearance of S_1 is equal to the rate of disappearance of S_2 in the system depicted in Figure 4.4. Mathematically, this condition is given by

$$\frac{dS_1}{dt} + \frac{dS_2}{dt} = 0. \tag{4.3}$$

When conservation relationships like Equation (4.3) are present in a biochemical network, there will be linearly dependent rows in the stoichiometric matrix. Therefore, the rank r will be less than the number of rows of the stoichiometric matrix. That is,

$$r = rank(\mathbf{N}) \leq m.$$

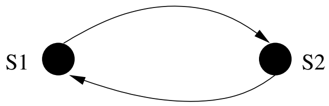


Fig. 4.4: Simple reaction network exhibiting conservation constraints.

Following the notation used by Reder [45] and Sauro and Ingalls [12], we can divide $\mathbf{N}$ into $\mathbf{N_R}$ and $\mathbf{N_0}$, the set of *independent* and *dependent* species, respectively. The concentrations of the independent metabolites, $\mathbf{N_R}$, can be used to calculate those of the dependent species $\mathbf{N_0}$. Thus, $\mathbf{N}$ can be expressed as

$$\mathbf{N} = \begin{bmatrix} \mathbf{N_R} \\ \mathbf{N_0} \end{bmatrix}. \tag{4.4}$$

Since $\mathbf{N_0}$ is a function of $\mathbf{N_R}$, there exists a matrix $\mathbf{L_0}$ satisfying

$$\mathbf{N_0} = \mathbf{L_0}\mathbf{N_R}. \tag{4.5}$$

This matrix is called the *link-zero* matrix. Equations (4.4) and (4.5) can be combined to yield

$$\mathbf{N} = \begin{bmatrix} \mathbf{N_R} \\ \mathbf{L_0}\mathbf{N_R} \end{bmatrix}. \tag{4.6}$$

Equation (4.6) can be further reduced by combining $\mathbf{L_0}$ with an identity matrix $\mathbf{I}$ and taking $\mathbf{N_R}$ as a common factor outside of the brackets, as shown below,

$$\mathbf{N} = \begin{bmatrix} \mathbf{I} \\ \mathbf{L_0} \end{bmatrix} \mathbf{N_R} = \mathbf{L}\mathbf{N_R}, \tag{4.7}$$

where $\mathbf{L} = [\mathbf{I} \;\; \mathbf{L_0}]^{\mathbf{T}}$ is called the *link* matrix. For systems in which conservation relationships do not exist, $\mathbf{N} = \mathbf{N_R}$, thus $\mathbf{L} = \mathbf{I}$.

The system equation can be written as $d\mathbf{S}/dt = \mathbf{N}\mathbf{v}$. It describes the time evolution of the reaction network and characterizes the kinetics of each individual reaction [12]. In this equation, $\mathbf{v}$ is the n-dimensional rate vector. Each rate in $\mathbf{v}$ is expressed as a function of the species concentrations. $\mathbf{S}$ is the species amounts vector. Similar to the stoichiometric matrix, $\mathbf{S}$ can be partitioned as $\mathbf{S}_i$ and $\mathbf{S}_d$, which are the independent and dependent species, respectively. We also call $\mathbf{S}$ the system state at time t.

System's equations can also be written in matrix form as

$$\frac{d\mathbf{S}}{dt} = \begin{bmatrix} d\mathbf{S_i}/dt \\ d\mathbf{S_d}/dt \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{L_0} \end{bmatrix} \mathbf{N_R}\mathbf{v}. \tag{4.8}$$

When Equation (4.8) is expanded, we obtain two separate equations

$$\frac{d\mathbf{S_i}}{dt} = \mathbf{N_R}\mathbf{v}, \tag{4.9}$$

$$\frac{d\mathbf{S_d}}{dt} = \mathbf{L_0}\mathbf{N_R}\mathbf{v}. \tag{4.10}$$

Combining Equations (4.9) and (4.10) we can re-express (4.10) as

$$\frac{d\mathbf{S_d}}{dt} = \mathbf{L_0}\frac{d\mathbf{S_i}}{dt}. \tag{4.11}$$

Integrating and rearranging (4.11) yields

$$S_d(t) - \mathbf{L_0}S_i(t) = S_d(0) - \mathbf{L_0}S_i(0), \tag{4.12}$$

which can be compacted to

$$[-\mathbf{L_0} \;\; \mathbf{I}] \begin{bmatrix} S_i(t) \\ S_d(t) \end{bmatrix} = \mathbf{T}. \tag{4.13}$$

In Equation (4.13), $\mathbf{T} = S_d(0) - \mathbf{L_0}S_i(0)$ is a constant vector which depends only on the initial conditions imposed of the system. If we let $[-\mathbf{L_0} \;\; \mathbf{I}] = \mathbf{\Gamma}$, it can be expressed as

$$\mathbf{\Gamma S} = \mathbf{T}. \tag{4.14}$$

$\mathbf{\Gamma}$ is called the *conservation* matrix. The rows of this matrix are related to the conserved cycles in the reaction network. Hence, the number of rows indicate the number of conserved species in the system. The nonzero elements of each row tell which species contribute to the corresponding conserved cycle.

There are several ways for computing the conservation matrix. Among existing methods are the right null space, Gauss-Jordan elimination, singular value decomposition (SVD), and reduced row echelon form [12]. Which method to choose depends on the network size and application. In this work we will address only one method to compute $\mathbf{\Gamma}$. Any method used to calculate the conservation matrix involves the computation of either $\mathbf{L_0}$ or $\mathbf{L}$, whether it is directly or not. For additional reading and information on the other methods refer to the work by Reder [45].

4.1.3 Computation of $\mathbf{\Gamma}$ Using the Null Space of N

Equation (4.5) can be expressed as

$$[-\mathbf{L_0} \;\; \mathbf{I}] \begin{bmatrix} \mathbf{N_R} \\ \mathbf{N_0} \end{bmatrix} = 0, \tag{4.15}$$

subsequently,

$$\mathbf{\Gamma N} = \mathbf{0}. \tag{4.16}$$

The conservation matrix $\mathbf{\Gamma}$ can thus be found by computing the null space of N. Equation (4.16) can be expressed as $\mathbf{N^T \Gamma^T} = \mathbf{0}$. Therefore, $\mathbf{\Gamma}$ can be computed as the right null space of $\mathbf{N^T}$. In a program like MATLAB this would be found by typing the command `transpose(null(transpose(N),'r'))`. It must be noted that in order to get correct output, $\mathbf{N}$ must be reordered as $[\mathbf{N_R} \;\; \mathbf{N_0}]^\mathbf{T}$. The link and link-zero matrices can then be extracted from $\mathbf{\Gamma}$. For in depth study and analysis of linear algebra theory, the reader is advised and encouraged to survey math college textbooks available in local or university libraries as well as in free sources available in the Internet. A very good introductory linear algebra textbook is “Introduction to Linear Algebra” by MIT professor Gilbert Strang [46].

4.2 Resolving Conservation Constraints in iSSA-MPDE

The Marginal Probability Density Evolution (MPDE) algorithm approximates the system using a *linear gaussian network* (LGN). This assumption relies on the following condition.

Conditional Independence: *the changes in any two species S_i, S_j must be statistically independent given the rest of the system's state at the start of each time-window.*

This condition can be easily verified using the covariance and information matrices. By assuming that the system is approximated by a LGN, the conservation constraints imposed on the network are violated. Therefore, the algorithm is no longer able to provide meaningful statistics that describe the intrinsic behavior of the system.

For example, consider the system shown in Figure 4.5. In this system promoter states are modeled as two separate species. P and P* represent unbound and bound promoter, respectively. The conservation law can be written as

$$P + P* = 1. \tag{4.17}$$

Equation (4.17) states that, at any given time t, the sum of the two promoter states must be constant. A simple way to solve the failure in MPDE for this system would be to generate P according to its statistics. P* is then determined using the conservation law in Equation (4.17).

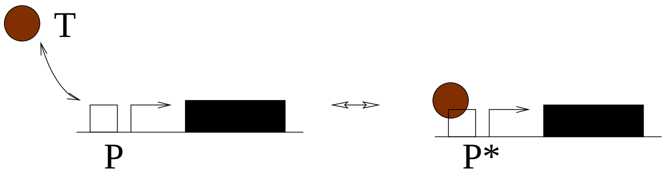


Fig. 4.5: Promoter states modeled as two separate species.

Though it is a very simple way to solve the issue, the conservation constraints are not always as clear and easy to determine. More complex and bigger systems might contain conservation relationships that are hard to see with the naked eye. In addition, it is not robust and might be tedious for genetic circuit designers.

The method introduced in Section 4.1.2 can be combined with MPDE to resolve the conservation constraints failure in a more robust and efficient way. Conservation laws can be determined automatically and resolved at run time. The refined algorithm version of MPDE, which is called MPDE-Conservation, is shown in Figure 4.6.

This enhanced MPDE algorithm was successfully tested using the VKBL circadian rhythm model [47] (see Appendix A.2). This circuit is known to oscillate with semi-random period. When multiple independent Gillespie's simulation stochastic algorithm (SSA) are averaged together, the oscillatory behavior is washed out. When raw MPDE is used, the statistics are not meaningful and the system does not oscillate. This happens because the conditional independence assumption is violated due to the presence of conservation constraints in the network. However, when the circuit is applied to the enhanced MPDE, the system oscillates and the statistics are meaningful. MPDE is substantially improved upon by identifying and resolving conservation constraints upfront. This new method can now process circuits in which conservation constraints due to species correlations are present without corrupting the results, thus making the algorithm much more robust, stable, and reliable.

4.3 MPDE and Principal Component Analysis (PCA)

Even after resolving linear conservation constraints in MPDE, there can appear other subtle (nonlinear) relationships among the chemical species that may or may not affect performance of MPDE. These types of correlations can be spotted at run time by computing the covariance matrix of the molecular species. The covariance matrix is a matrix whose (i, j)th element is the covariance between the ith and jth species when $i \neq j$, and the variance of the jth species when $i = j$. This matrix is ideally diagonal when no correlations are present in the variables. Once the covariance matrix is computed, it can be used by the

1. Identify conservation constraints in the system.
2. At time t, each independent species x_i is represented by a *mean value* and a *variance.*
3. Use equation (4.12) to compute S_d from S_i.
4. Repeat the following steps N times:
    (a) For each species x_i, generate a random value based on its mean, variance and statistical type.
    (b) Run the SSA algorithm until τ seconds has elapsed. (τ is the time-window).
    (c) Record the mean and variance of each x_i at the end of the time-window.
5. Continue until the desired simulation time is reached.

Fig. 4.6: MPDE algorithm with conservation constraints resolution.

PCA algorithm to resolve correlations among the chemical species. It is very interesting to note that linear relationships between the variables can be seen from the covariance matrix when there are zeros in the elements of the main diagonal [42]. However, it is preferable to resolve linear relationships using conservation analysis before applying PCA for performance reasons as will be explained later.

Principal component analysis is a method that aims to reduce the dimensionality of a data set consisting of interrelated variables. In order to achieve this, PCA transforms the data space into an smaller number of effective variables, called *principal components*, that are uncorrelated and retain most of the variations present in the original variables. Adopting the notation used by Jolliffe [42], the first step in PCA is to find a linear function of the form $\boldsymbol{\alpha}_1^T\mathbf{x}$ of maximum variance, where $\mathbf{x}$ is a vector of m random variables (chemical species) and $\boldsymbol{\alpha}_1$ is a vector of m constants $\alpha_{11}, \alpha_{12,\ldots,\alpha_{1m}}$, such that

$$\boldsymbol{\alpha}_1^T\mathbf{x} = \sum_{j=1}^{m} \alpha_{1j} x_j.$$

Then, look for a second function $\boldsymbol{\alpha}_2^T\mathbf{x}$ having maximum variance and uncorrelated to $\boldsymbol{\alpha}_1^T\mathbf{x}$. Do this until the kth function $\boldsymbol{\alpha}_k^T\mathbf{x}$ is found such that it is uncorrelated with $\boldsymbol{\alpha}_1^T\mathbf{x}$,

$\boldsymbol{\alpha}_2^T\mathbf{x}$,..., $\boldsymbol{\alpha}_{k-1}^T\mathbf{x}$. The kth function $\boldsymbol{\alpha}_k^T\mathbf{x}$ is the kth principal component (PC). Even though up to m principal components can be found, it is hoped, in general, that most of the variation will be retained by $p \ll m$ PCs.

Having defined the principal components, it is now when knowing the covariance matrix aids in computing the PCs. Let Σ denote the covariance matrix of the data set. it turns out that the kth PC is given by

$$z_k = \boldsymbol{\alpha}_k^T\mathbf{x},$$

where $\boldsymbol{\alpha_k}$ is an eigenvector of Σ corresponding to its kth largest eigenvalue. Define $\mathbf{z}$ as the vector whose kth element is z_k. Then

$$\mathbf{z} = \mathbf{A}^T\mathbf{x},$$

where $\mathbf{A}$ is the orthogonal matrix whose kth column, $\boldsymbol{\alpha}_k$, is the kth eigenvector of Σ corresponding to its kth eigenvalue.

PCA can be easily implemented in mathematical engines like MATLAB. For instance, the following steps can be taken to compute the principal components in MATLAB, assuming X is a matrix whose columns are the data variables.

```
C = cov(X);
[V L] = eig(C);
% take only the first few vectors of 'V' correponding to the largest
% eigenvalues 'L' and put them as columns in matrix 'A'.
Z = A'X;
```

PCA can be potentially used within MPDE to resolve correlations between chemical species in a similar way to conservation constraints analysis. The main difference is that the set of “independent” species would be the principal components (PCs) and those would be used to run MPDE at the beginning of every time increment. Then, at the end of the time increment, data is transformed back into its original state to have corrected, uncorrelated data. For more information about PCA, the reader is encouraged to read the book of Jolliffe titled *Principal Component Analysis* [42]. This is a great resource containing detailed explanations and lots of references on the theory and application of PCA.

PCA is a good "general" method to uncorrelate data as well as to reduce (or compress if you will) dimensionality. However, whenever possible, conservation constraint analysis is preferred because its simplicity and robustness as well. Conservation constraint analysis is less computationally complex, thus faster than PCA. If after applying MPDE-conservation there still remain correlations among the species that cause MPDE to fail, then PCA would come in handy to resolve the bad correlations. The extent to which conservation analysis is able to keep MPDE from breaking linearly correlated species can be "measured" by computing the cross-correlation coefficients at run time. These coefficients, also known as Pearson's correlation, are the most familiar measure of independence between two data vectors. They can be arranged in a *correlation matrix* as shown below in Σ_1 through Σ_5. The correlation matrix can be expressed as a function of the species' covariances in the following equation

$$Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_x \sigma_y}.$$

$$\Sigma_1 = \begin{pmatrix} 1.00 & -0.09 & -1.00 & 0.53 \\ -0.09 & 1.00 & 0.09 & -0.89 \\ -1.00 & 0.09 & 1.00 & -0.53 \\ 0.53 & -0.89 & -0.53 & 1.00 \end{pmatrix}$$

$$\Sigma_2 = \begin{pmatrix} 1.00 & 0.25 & -1.00 & 0.32 \\ 0.25 & 1.00 & -0.25 & -0.83 \\ -1.00 & -0.25 & 1.00 & -0.32 \\ 0.32 & -0.83 & -0.32 & 1.00 \end{pmatrix}$$

$$\Sigma_3 = \begin{pmatrix} 1.00 & 0.30 & -1.00 & -0.03 \\ 0.30 & 1.00 & -0.30 & -0.96 \\ -1.00 & -0.30 & 1.00 & 0.03 \\ -0.03 & -0.96 & 0.03 & 1.00 \end{pmatrix}$$

$$\Sigma_4 = \begin{pmatrix} 1.00 & -0.09 & -1.00 & 0.67 \\ -0.09 & 1.00 & 0.09 & -0.79 \\ -1.00 & 0.09 & 1.00 & -0.67 \\ 0.67 & -0.79 & -0.67 & 1.00 \end{pmatrix}$$

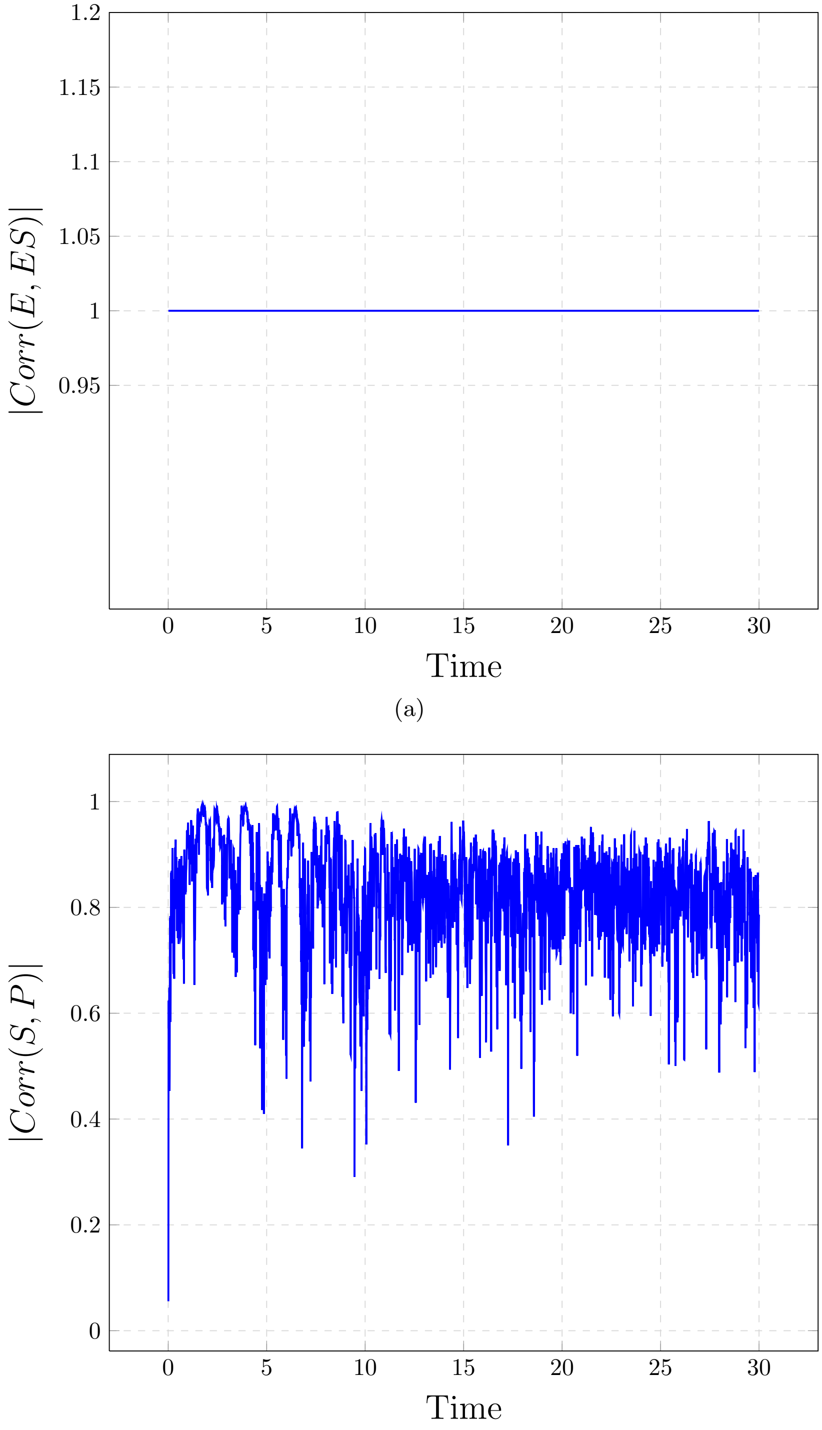
$$\Sigma_5 = \begin{pmatrix} 1.00 & -0.04 & -1.00 & 0.37 \\ -0.04 & 1.00 & 0.04 & -0.94 \\ -1.00 & 0.04 & 1.00 & -0.37 \\ 0.37 & -0.94 & -0.37 & 1.00 \end{pmatrix}$$

$$\Sigma_{MPDE} = \begin{pmatrix} 1.00 & 0.05 & -0.05 & 0.07 \\ 0.05 & 1.00 & -0.76 & 0.80 \\ -0.05 & -0.76 & 1.00 & -0.87 \\ 0.07 & 0.80 & -0.87 & 1.00 \end{pmatrix}$$

The correlation matrix is symmetric and can be thought of as a normalization of the covariance matrix where values are constrained to the interval $[-1, 1]$. The cross-correlation coefficient will be equal to 1 for a positive linear relationship (when both quantities increase together); for negative linear relationships the correlation will be -1, while 0 for totally uncorrelated data. Thus, for a set of completely independent vectors the correlation matrix will be equal to the identity matrix. Matrices Σ_1 through Σ_5 show how MPDE-conservation is able to maintain the correlated species intact without breaking any conservation constraint in the system. These species are (E,ES) and (S,P) in entries (1,3) and (2,4), respectively. On the other hand, Σ_{MPDE}, MPDE correlation matrix, shows that MPDE breaks the negative linear relationship between species E and ES (entries (1,3) and (3,1)).

Figure 4.7(a) depicts the time evolution of the absolute value of the cross-correlation between species (E,ES). MPDE-conservation successfully maintains this conservation law throughout the entire simulation time. Next, Figure 4.7(b) plots the correlations of species (S,P), which are the other set of species linearly related in the system. Though not purely linear, MPDE-conservation is not only able to identify it, but also retain it at all times. There are other minor correlations among other pair species, as shown in Figure 4.8(a) and Figure 4.8(b). However, they do not affect the capability of MPDE-conservation to deliver the expected results. Figure 4.9(a) depicts the time evolution of correlation values for species (E,S), while Figure 4.9(b) shows the corresponding plot for species (S,ES), indicating that the relationship between this set of species is almost purely uncorrelated. Finally, we plotted the average of the cross-correlation coefficients for every pair of species in Figure 4.10. It shows that the linear correlations are consistent for those species that the conservation analysis method identified as independent and dependent. The two highest values (1.0 and 0.82) correspond to the linearly related species, while the rest correspond to the "independent" set of species.

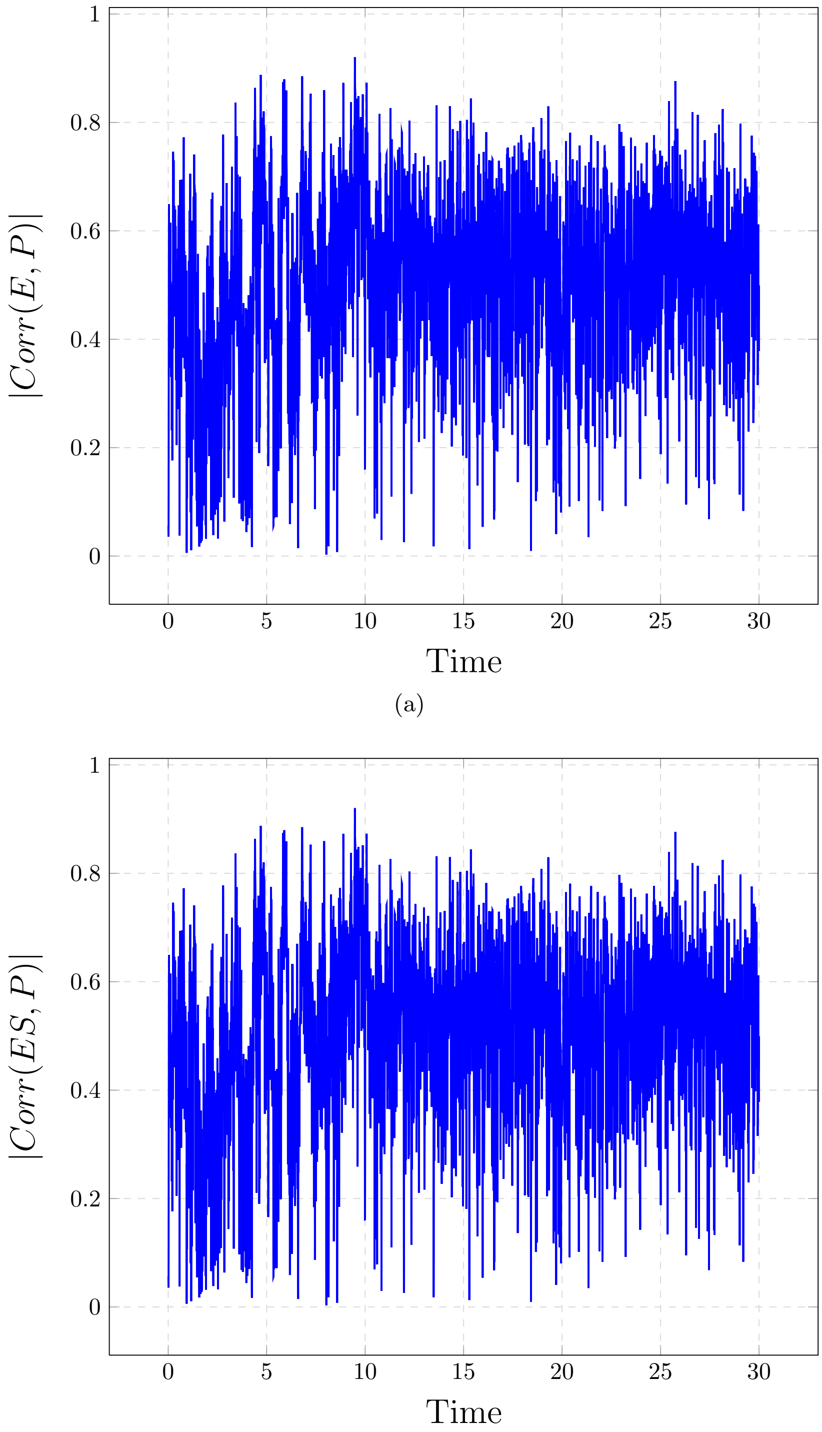
To further explain and understand PCA and its role along side MPDE, let us show the following examples using the Michaelis-Menten system. Simulation data of a stochastic run of the Michaelis-Menten enzymatic system was used to show how linearly correlated two of



(a)

(b)

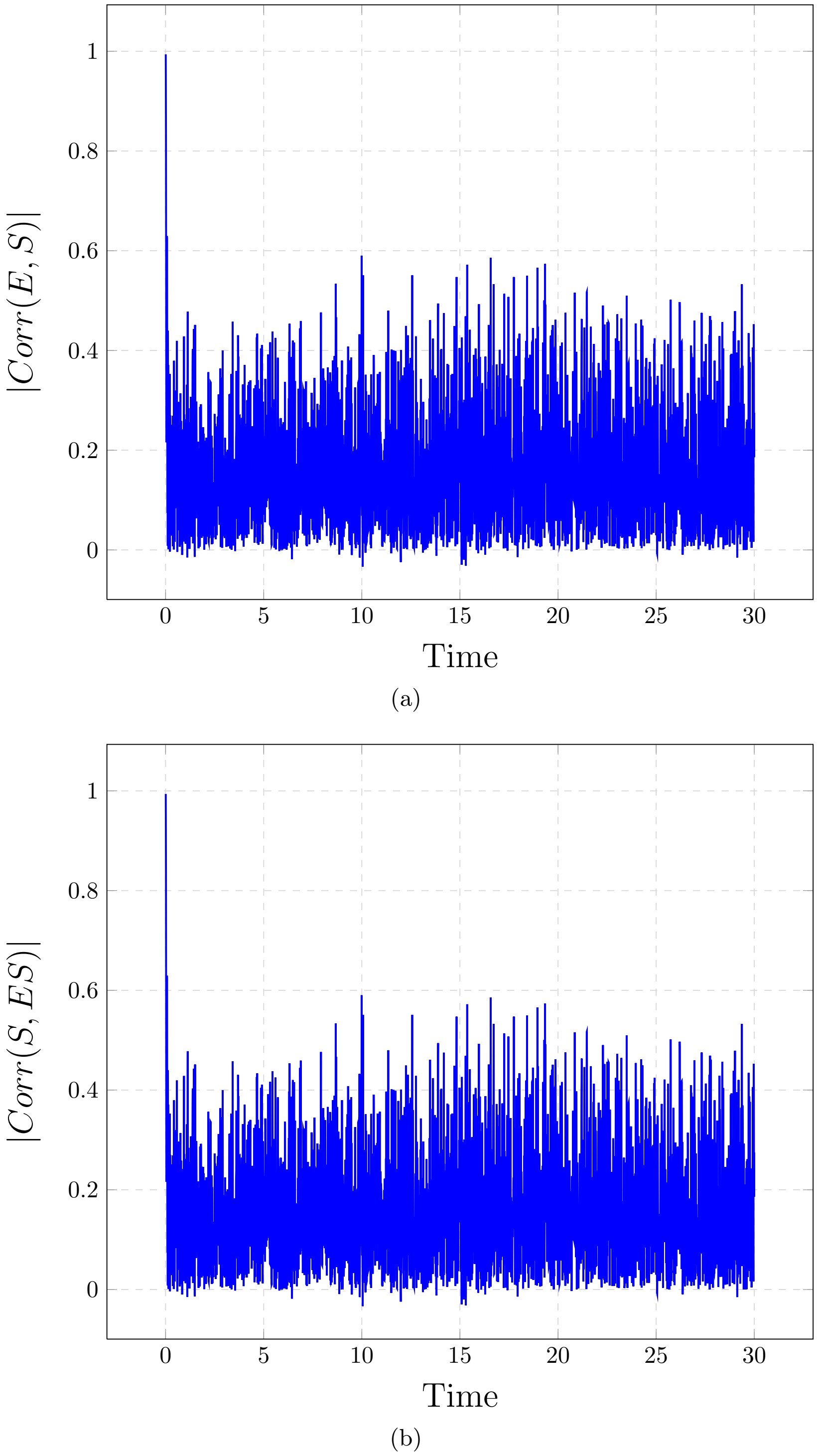
Fig. 4.7: Illustration of the time evolution of the cross-correlation between pair of species of the Michaelis-Menten system: (a) Cross-correlation between species E and ES, (b) Cross-correlation between species S and P.



(a)

(b)

Fig. 4.8: Illustration of the time evolution of the cross-correlation between pair of species of the Michaelis-Menten system: (a) Cross-correlation between species E and P, (b) Cross-correlation between species ES and P.



(a)

(b)

Fig. 4.9: Illustration of the time evolution of the cross-correlation between pair of species of the Michaelis-Menten system: (a) Cross-correlation between species E and S, (b) Cross-correlation between species S and ES.

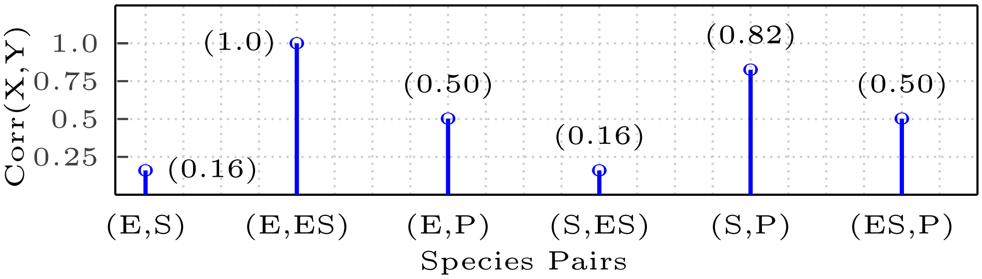


Fig. 4.10: This picture depicts the simulation time average of the absolute value of species cross-correlations for the Michaelis-Menten System.

the species are and how the PCs show similar variation to that of the original data. First, consider the covariance matrix, Σ, below. This symmetric matrix shows the variances of the species along the main diagonal, whereas the non-diagonal elements show the covariances between pairs of species. For illustration, the eigenvalues and eigenvectors, Λ and A, respectively, are shown below. The principal components are computed using the last two eigenvectors in A, which are the ones that correspond to the largest two eigenvalues in Λ. Figure 4.11 plots the time evolution of the trace of the covariance matrices computed at the end of each time increment in the MPDE-conservation algorithm. The fact that this figure settles at a small value indicates that the individual variances of the species are very small. Figure 4.12 depicts a scatter plot of the linearly correlated species. After applying PCA, the principal components are plotted in Figure 4.13(a) which resemble the shape of species S and ES of the original data shown in Figure 4.13(b). These findings reinforce the idea that PCA can indeed be used as an alternate method in case persistent correlations remain after applying MPDE-conservation.

$$\Sigma = \begin{pmatrix} 16.66 & -16.66 & 88.30 & -71.64 \\ -16.66 & 16.66 & -88.30 & 71.64 \\ 88.30 & -88.30 & 732.05 & -643.75 \\ -71.64 & 71.64 & -643.75 & 572.11 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} -9.55 \times 10^{-14} & 0 & 0 & 0 \\ 0 & 3.31 \times 10^{-16} & 0 & 0 \\ 0 & 0 & 16.65 & 0 \\ 0 & 0 & 0 & 13.21 \times 10^{2} \end{pmatrix}$$

$$A = \begin{pmatrix} 0.32 & 0.71 & -0.63 & -0.08 \\ -0.32 & 0.71 & 0.63 & 0.08 \\ -0.63 & -0.00 & -0.21 & -0.74 \\ -0.63 & 0.00 & -0.41 & 0.65 \end{pmatrix}$$

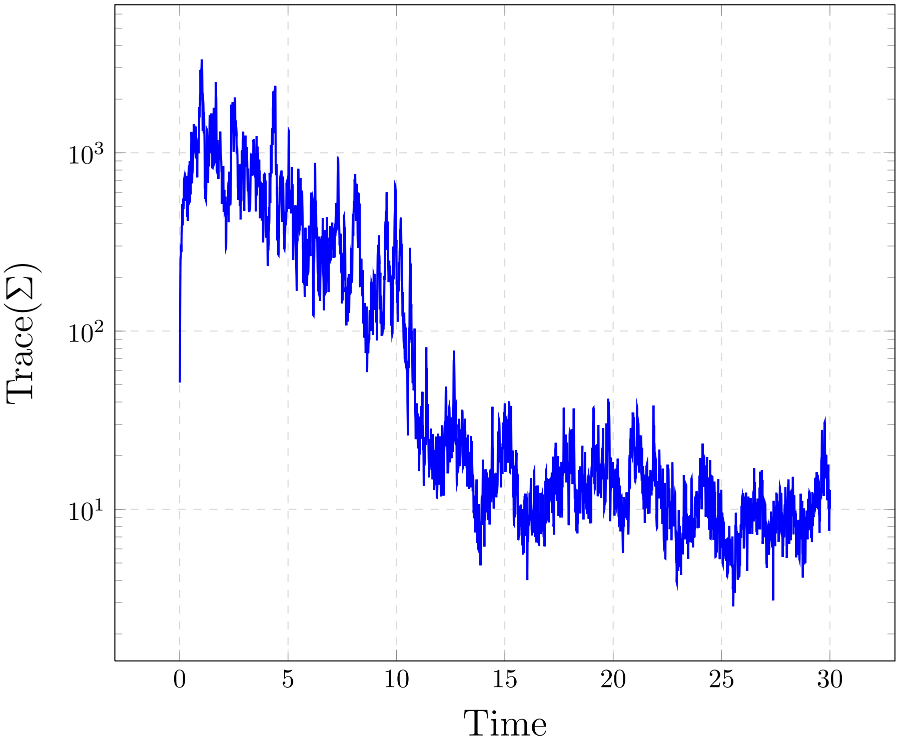
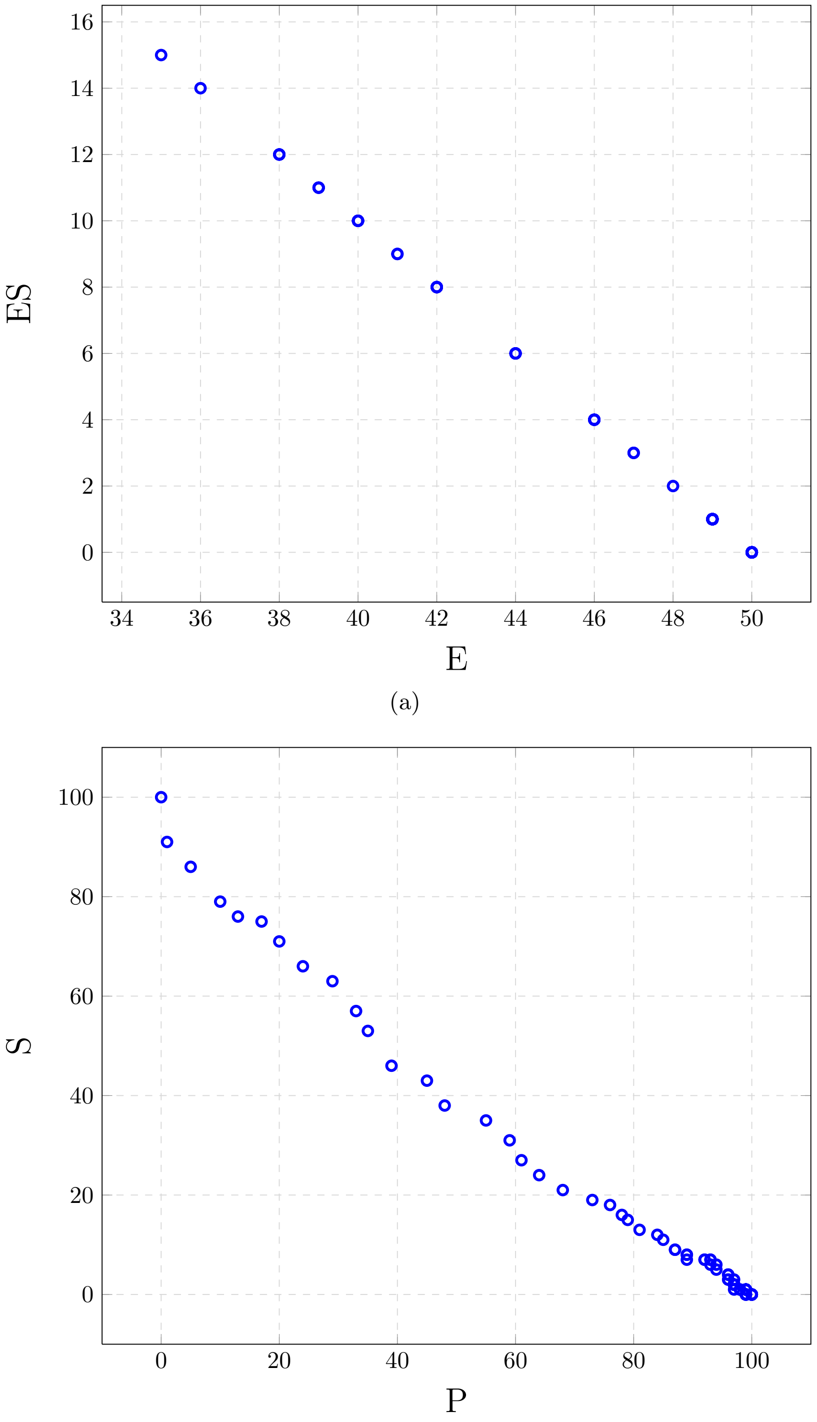


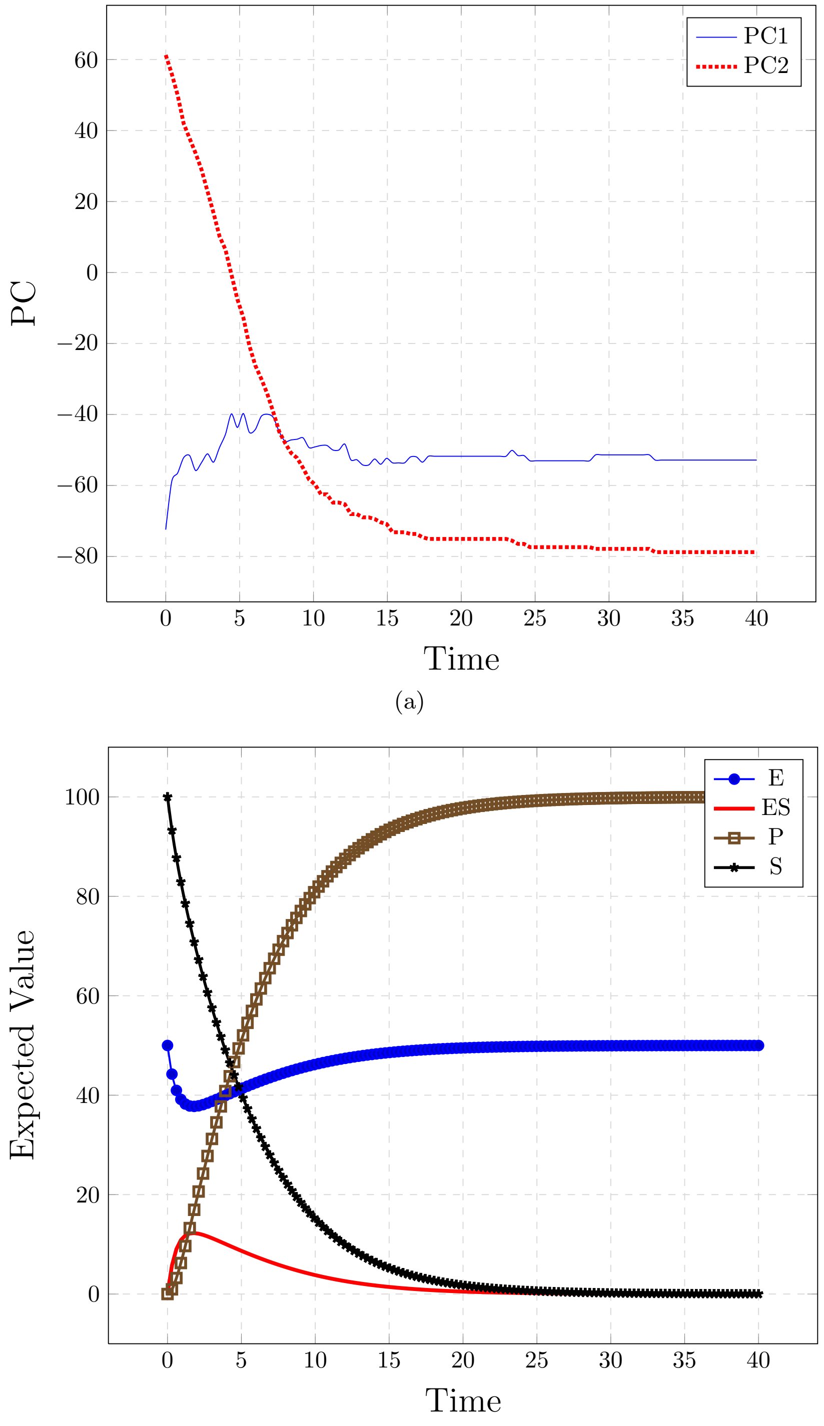
Fig. 4.11: This figure depicts the time evolution of the trace of the covariance matrix for a run of the Michaelis-Menten model with MPDE-conservation.



(a)

(b)

Fig. 4.12: This plot depicts the linear relationship or "correlation" of the species in the Michaelis-Menten reaction system. This information is easily obtained from the covariance matrix, Σ, above. (a) Scatter plot of the correlation between species E and ES, (b) Scatter plot showing the correlation between species P and S.



(a)

(b)

Fig. 4.13: Comparison between the simulation data of the Michaelis-Menten reaction system and the Principal Components obtained from PCA showing that the variation of the PCs is very similar to the original data from the ODE simulation. (a) Plot of the Principal Components (PCs) against time, (b) ODE simulation plot of the Michaelis-Menten system.

Chapter 5
Results

The Marginal Probability Density Evolution (MPDE) has been implemented within iBioSim and MPDE with conservation constraint resolution has been implemented in C/C++. Several example models have been applied to test the performance and accuracy of MPDE to compare it with that of SSA. Models such as the VKBL circadian rhythm [47], [1] and the Michaelis-Menten enzymatic reaction system [33], were taken as starting models because of their detailed description and documentation in the literature. The remaining of this chapter is organized as follows: all simulation results pertaining to the Michaelis-Menten enzymatic reaction network is presented in Section 5.1 and the aforementioned VKBL circadian rhythm is explored in Section 5.2.

5.1 Michaelis-Menten

The first model considered is the Michaelis-Menten reaction system. The chemical reaction network for this system is described in Appendix A.1 and depicted in Figure 5.1. This model describes the velocity of enzymatic reactions by relating the rate with the concentration of a substrate. Figure 3.2 on Page 25 shows the ODE simulation results for the Michaelis-Menten system. For a simple system like this, ODE models work well. However, it has been widely argued, as stated in Chapter 3, that ODE models assume that the system varies deterministically and continuously. This assumption might be valid when there are large amounts of molecules and the cell volume is well-stirred. The fact is that most systems are discrete and stochastic, meaning that there is a small amount of molecules and chemical reactions occur at random. Therefore, stochastic models are required to better capture the true behavior of biochemical networks.

[1] The SBML file and network description for this model can be found in biomodels database under ID BIOMD0000000101 (http://www.ebi.ac.uk/biomodels-main/publmodels).

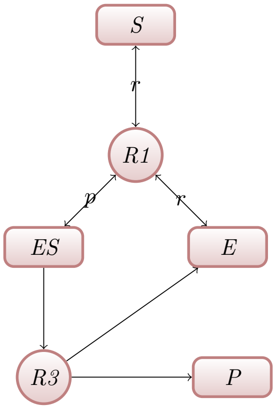


Fig. 5.1: Michaelis-Menten schematic diagram as drawn in iBioSim. Species are represented by curved rectangles while reactions are drawn as small circles labeled R_j. Arrows leaving from a species and arriving in a reaction R_j indicate that species R_j is a reactant in this particular reaction. Arrows arriving at a species S_i indicate that reaction R_j affects species S_i. Also, reactions are affected by modifier species. This case is represented by a line connecting the modifier species and the reaction it participates in.

Figure 5.2(a) shows the results after simulating the Michaelis-Menten system using Gillespie's Stochastic Simulation Algorithm (SSA). One can notice that the behavior is very similar to that of the ODE results. Indeed, for a very large amount of molecules these two simulations will be nearly identical. SSA simulations produce a single path or sample from the probability density function described by the chemical master equation (CME). Very often designers are interested in answering two fundamental questions: what is the "typical" behavior of the systems and how robust is it? A simple and intuitive approach is to take the average of multiple independent runs of SSA as shown in Figure 5.2(b). For simple systems like this averaging yields accurate results. However, most interesting biological systems in nature, as well as synthetically engineered, exhibit a high degree of stochasticity. For highly stochastic models simply aggregating tends to distort the intrinsic behavior of the biological system and this effect will be shown in the subsequent sections.

This model was also simulated using MPDE and the result is shown in Figure 5.3(a). Though not very accurate, MPDE is able to capture the intrinsic behavior of the enzymatic network. The inaccuracies in the MPDE simulations are due violations of some conservation constraints.

To illustrate the effect of conservation constraints on MPDE, let us first consider the enzymatic reaction system of Michaelis-Menten in Appendix A.1. This model contain a couple of weakly correlated species and the conservation law is given by $[E]+[ES]=K$ and $[S]+[P]=K+[E]$, where K is some initial condition (molecule amount). When MPDE is applied to this model the result is close to the expected behavior, but it deviates a little and is not as smooth and clean as a single SSA run or average of multiple SSA runs, as shown in Figure 5.3(a). Broken conservation laws force MPDE to deviate from the true behavior of the system. However, these constraints are resolved when the refined version of MPDE, MPDE-conservation, is applied to the system. Figure 5.3(b) depicts the results corresponding to MPDE-conservation, yielding a smooth path that is true to the expected behavior of the enzymatic network.

5.2 VKBL Circadian Rhythm

The VKBL oscillator is a minimal model of a circadian rhythm based on positive and negative feedback networks [47]. It is composed primarily of two genes, an activator A and a repressor R. A acts as a positive element in transcription by binding to the A and R promoters to increase transcription rates. On the other hand, repressor R acts as a negative element by inhibiting the activator. The schematic for this circuit is shown in Figure 5.4 and the corresponding reaction network is discussed in Appendix A.2.

When simulated with both deterministic (ODE) and stochastic (SSA) models, this system exhibits an oscillatory behavior (See Vilar *et al.* [47] for ODE simulations). In the deterministic model[2], every oscillation is identical to the previous one, whereas the stochastic model shows variability both in the number of molecules and the period of oscillation, as seen in Figure 5.5(a). These variations correspond to inherent fluctuations of the biochemical network.

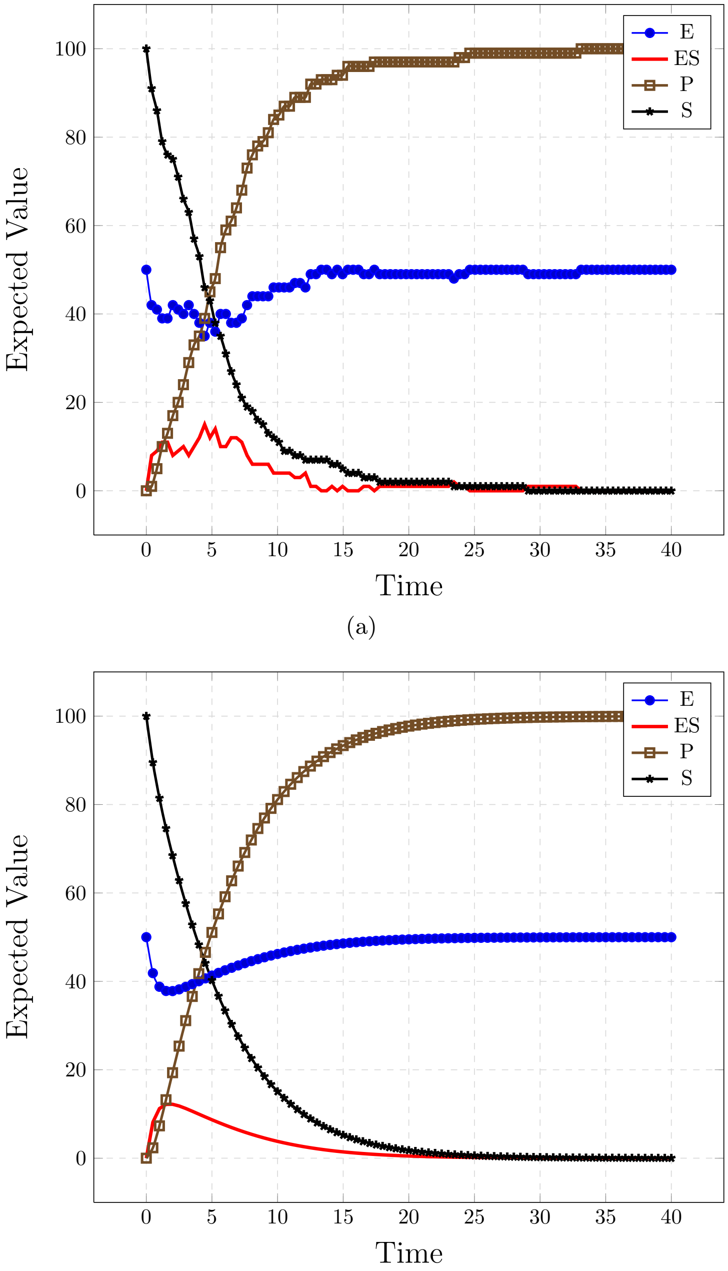
Under certain conditions or, more specifically, for some values of parameters, both the stochastic and deterministic approaches produce similar results. However, Vilar *et al.* [47] have found that parameters that indicate a stable steady state using the deterministic

[2] All deterministic simulations referred to in this section have been referenced to the work of Vilar *et al.* [47].

approach continue to produce sustained oscillations when simulated with SSA. Therefore, the presence of noise can change the behavior of the system revealing marked qualitative differences that cannot be observed by deterministic means [47].

As pointed out before, single SSA runs cannot reveal the clean average "signal" hidden under noise. The common approach to filter out the noise component is by averaging multiple independent stochastic simulations. However, as shown in Figure 5.5(b), the method of averaging may wash out the true behavior of highly stochastic systems. The proposed solution is to use MPDE [11]. Nevertheless, even MPDE fails when there are conservation constraints in a biochemical network like the VKBL oscillator. Figure 5.6(a) illustrates the effect of conservation constraints violation in MPDE, where all oscillations are concealed but just the first.

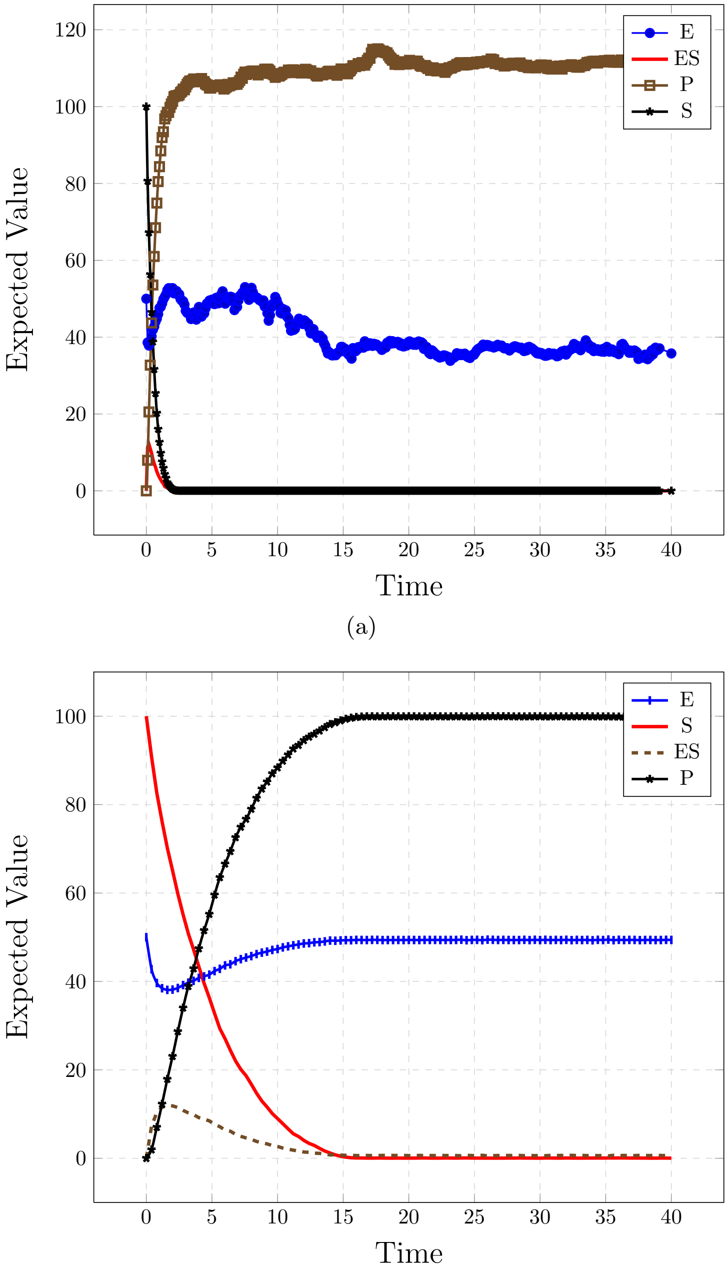
The core of MPDE relies on the assumption that any pair of species in the system are conditionally independent given the rest of the system. Nonetheless, if dependencies appear among the species, MPDE may perform poorly and even yield incorrect results. The matter of fact is that most interesting biological systems in nature as well as man made contain tightly correlated species. In addition, an increasing number of computational methods use forms of abstraction to accelerate the computation time by reducing the effective number of chemical reactions. When systems are abstracted in this way, dependencies may be introduced [43]. Therefore, MPDE is not attractive to simulate systems that have been abstracted [48, 49]. Hence, a method is required to correct errors caused by MPDE when conservation constraints appear. The technique presented in Chapter 4 successfully resolves these types of errors by identifying dependent species from independent species. After moieties have been properly separated, MPDE is run only for the independent species, which can then be used to compute the states of the dependent species. Figure 5.6(b) shows how MPDE-conservation successfully maintains conservation laws intact, producing the expected behavior of this oscillatory network.



(a)

(b)

Fig. 5.2: Results of simulating the Michaelis-Menten enzymatic reaction system using SSA. (a) Depiction of a single SSA path, (b) Shows the result of averaging 100 independent SSA simulation runs. The parameters used to produce these simulations are given in Appendix A.1. Simulated using iBioSim.



(a)

(b)

Fig. 5.3: Results of simulating the Michaelis-Menten enzymatic reaction using MPDE. (a) Simulation obtained using MPDE, (b) Results of simulating the model with MPDE-conservation. The parameters used to produce this figures were: $\tau = 0.1$ and $runs = 200$.

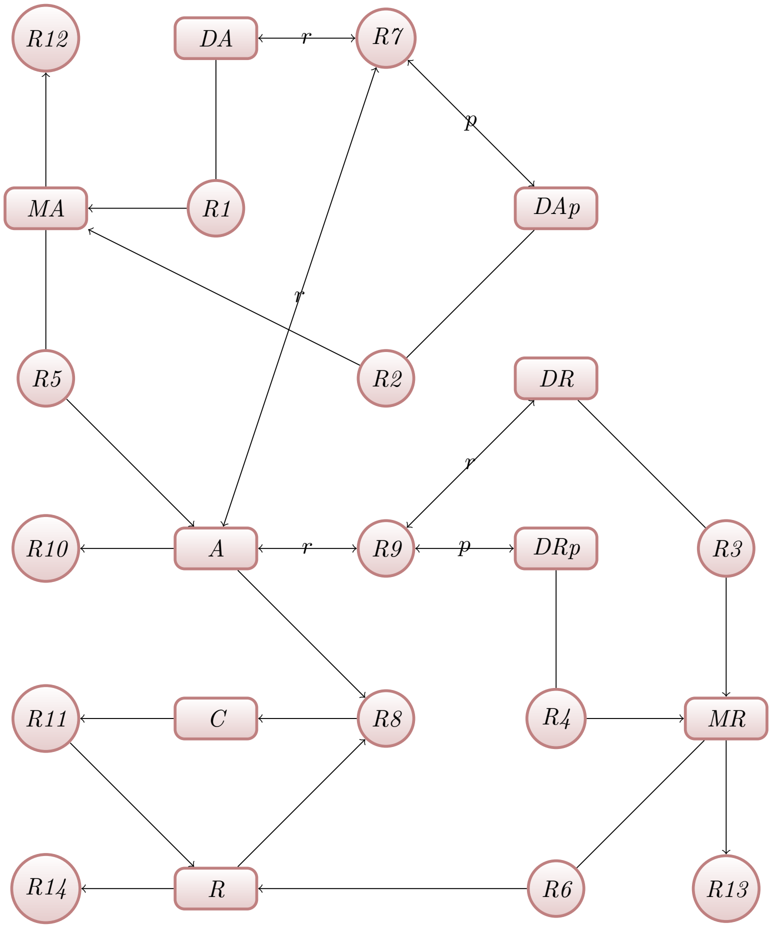
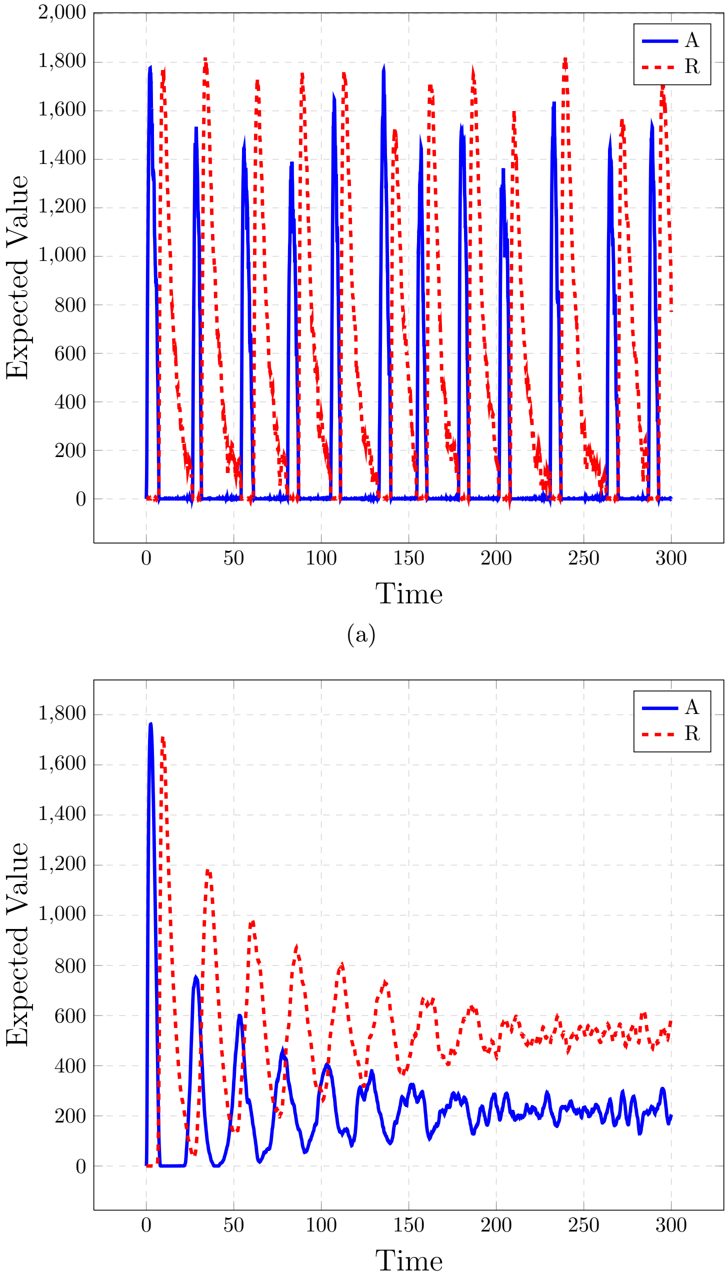


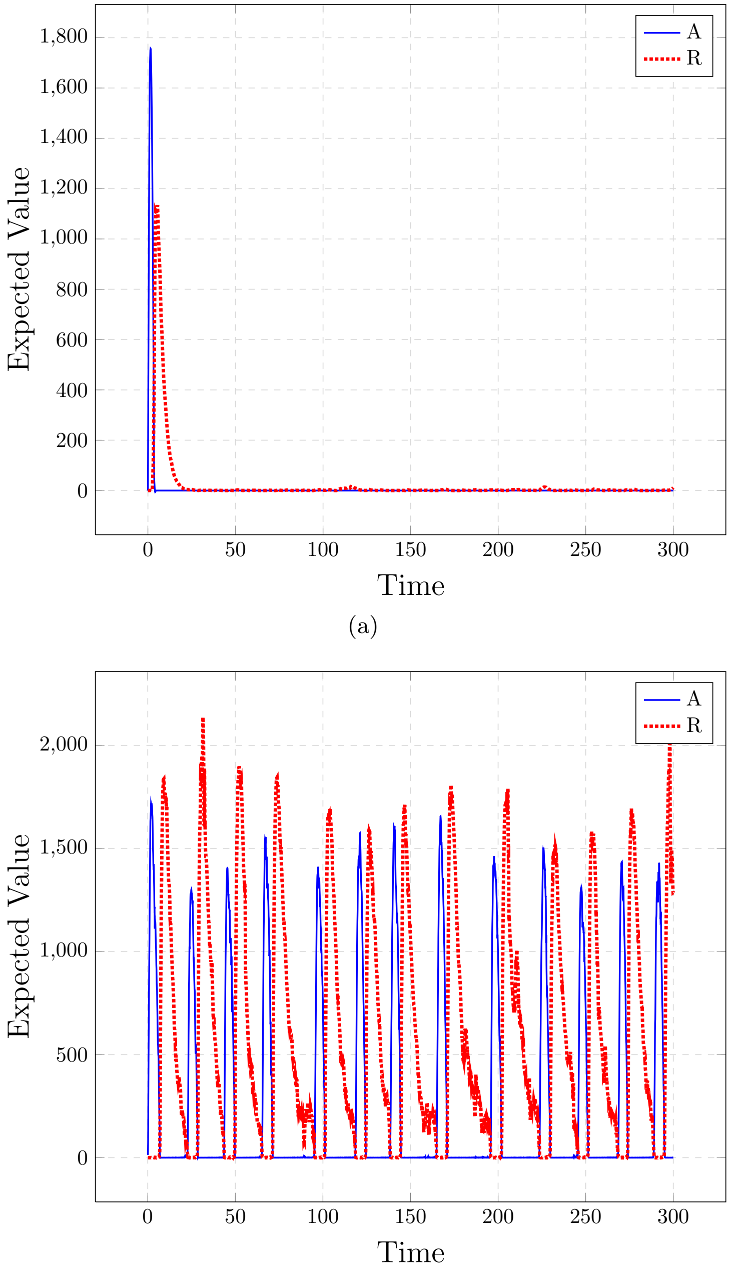
Fig. 5.4: VKBL schematic diagram as drawn in iBioSim. Species are represented by curved rectangles while reactions are drawn as small circles labeled R_j. Arrows leaving from a species and arriving in a reaction R_j indicate that species R_j is a reactant in this particular reaction. Arrows arriving at a species S_i indicate that reaction R_j affects species S_i. Also, reactions are affected by modifier species. This case is represented by a line connecting the modifier species and the reaction it participates in.



(a)

(b)

Fig. 5.5: Stochastic simulations of the VKBL circadian rhythm: (a) Single SSA run for 300 seconds, (b) Average of 100 SSA runs illustrating how this approach can wash out the expected behavior of highly stochastic and/or oscillatory networks. These simulations were performed using the VKBL model from the biomodels database in iBioSim.



(a)

(b)

Fig. 5.6: Stochastic simulations using MPDE and MPDE-conservation. (a) MPDE results illustrating the effects of conservation constraints. Conservation constraints impose biochemical laws among the chemical species that are broken by MPDE. (b) Results of MPDE-conservation showing successful results when conservation laws are taken into account in MPDE.

Chapter 6

Discussion

While the results show that MPDE is a promising algorithm to simulate genetic circuits, there are areas in which in can be further improved. First, MPDE uses SSA as its core algorithm. Modifying MPDE to use more efficient methods such as tau-leaping would increase its computational efficiency. Efficient SSA variants often contain restrictions that would be passed on to MPDE. It would be beneficial to investigate ways to combine different simulation methods at run time to accelerate MPDE computational time. In addition, making the time increment adaptive might increase the efficiency of MPDE as well.

Second, MPDE relies on the assumption that species are conditionally independent. This independence approximation has proven to be accurate for detailed reaction networks. However, when abstraction methods are employed, highly correlated species may appear. Further research should be done to lift the conditional independence restriction and find a method to generate more accurate molecule counts for every species without violating conservation constraints. In the same line, the way the conservation constraints identification algorithm is currently implemented can be written in a more efficient manner. Presently, both MPDE and MPDE-Conservation are implemented separately. These two methods can be combined such that if conservation laws do not exist in the system, then MPDE is run without having to verify and periodically check that conservation constraints are met. On the other hand, if there are conservation relationships in the reaction network, then MPDE-Conservation is used for the simulation.

In addition to linear relationships, there may appear other subtle correlations among the molecular species that are not detected by conservation constraint analysis. These correlations can be identified by computing the covariance matrix at the beginning of each time increment in MPDE. If there exists any correlations, Principal Component Analysis

(PCA) can be used to transform the space into an effective number of uncorrelated species, the principal components, that hold most of the variations of the system. The principal components are used to run the simulation until the end of the time increment, at which time the space is transformed back into its original species, thus having corrected the correlations. The drawback with this method, as opposed to conservation analysis, is the computational burden of having to find the covariance matrix and running PCA for every time increment. Whereas conservation analysis is done only once, at the beginning of the simulation, and uses a simple equation to compute the dependent species from the independent species, thus making it the preferred method. All in all, PCA would be great tool to be used if, after doing conservation analysis, there still remain correlations that make even MPDE-conservation fail.

Furthermore, there are genetic circuits that can exhibit bi-stability, like the toggle switch presented by Wilhelm [41], or multi-stability. For such systems, MPDE is only able to follow a single path, which turns out to be the most likely path. It would be useful to extend MPDE's capability to identify bifurcations and compute the probability of each possible path. Finally, the capabilities of MPDE can be exploited and further improved if it is implemented in a tool such as iBioSim. [1]

To finalize this work, there are two major limitations about MPDE with conservation constraint resolution listed below. The first one, simulation time, is inherent to the structure of the algorithm and very little can be done about it. The second limitation can be further explored by using PCA to detect and correct nonlinear correlations among the species. Nonetheless, it will add some overhead time, thus making the algorithm slower.

- Slightly slower than MPDE and much more slower than SSA;
- Only linear relationships can be corrected.

[1] See http://www.async.ece.utah.edu/iBioSim/.

References

[1] T. Danino, O. Mondragón-Palomino, L. Tsimring, and J. Hasty, "A synchronized quorum of genetic clocks," *Nature*, vol. 463, no. 7279, pp. 326–30, Jan. 2010. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2838179\&tool=pmcentrez\&rendertype=abstract

[2] H. Kuwahara and I. Mura, "An efficient and exact stochastic simulation method to analyze rare events in biochemical systems," *Journal of Chemical Physics*, vol. 129, no. 16, 2008.

[3] C. J. Myers, *Engineering Genetic Circuits*, Salt Lake City, UT, 2009. [Online]. Available: http://books.google.com/books?hl=en\&amp;lr=\&amp;id=UoCq4FeTeHkC\&amp;oi=fnd\&amp;pg=PR13\&amp;dq=Engineering+Genetic+Circuits\&amp;ots=tNA_R1ab7V\&amp;sig=y7GSh8yfyj8B9YhtBF9mMR2rz_U

[4] B. D. Fett, "Synthesizing Stochasticity in Biochemical Systems," Ph.D. dissertation, University of Minnesota, MN, 2010.

[5] N.-p. Nguyen, C. Myers, H. Kuwahara, C. Winstead, and J. Keener, "Design and analysis of a robust genetic Muller C-element." *Journal of Theoretical Biology*, vol. 264, no. 2, pp. 174–87, May 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19914258

[6] D. T. Gillespie, "Stochastic simulation of chemical kinetics," *Annual review of physical chemistry*, 2007. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.147.8267\&amp;rep=rep1\&amp;type=pdf

[7] H. E. Samad, M. Khammash, H. El Samad, L. Petzold, and D. Gillespie, "Stochastic modelling of gene regulatory networks," *International Journal of Robust and Nonlinear Control*, vol. 15, no. 15, pp. 691–711, Oct. 2005. [Online]. Available: http://doi.wiley.com/10.1002/rnc.1018http://onlinelibrary.wiley.com/doi/10.1002/rnc.1018/abstract

[8] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001.

[9] J. Hasty, F. Isaacs, M. Dolnik, D. McMillen, and J. J. Collins, "Designer gene networks: Towards fundamental cellular control," pp. 207–220, 2001. [Online]. Available: http://scitation.aip.org/getpdf/servlet/GetPDFServlet?filetype=pdf\&id=CHAOEH000011000001000207000001\&idtype=cvips\&prog=normal\&doi=10.1063/1.1345702

[10] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in Escherichia coli." *Nature*, vol. 403, no. 6767, pp. 339–42, Jan. 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10659857

[11] C. Winstead, C. Madsen, and C. Myers, “iSSA : An Incremental Stochastic Simulation Algorithm for genetic circuits,” in *International Symposium on Circuits and Systems (ISCAS).* Paris, France: Proceedings of 2010 IEEE, 2010, pp. 553–556. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5537539\&tag=1

[12] H. M. Sauro and B. Ingalls, “Conservation analysis in biochemical networks: computational issues for software writers.” *Biophysical chemistry*, vol. 109, no. 1, pp. 1–15, Apr. 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15059656

[13] S. S. C. Hilgetag and J. H. W. D. A. Fell, “Reaction routes in biochemical reaction systems: Algebraic properties, validated calculation procedure and example from nucleotide metabolism,” vol. 181, pp. 153–181, 2002.

[14] J.-h. S. Hofmeyr, “Metabolic control analysis in a nutshell,” *Differentiation*, no. ii, pp. 291–300.

[15] D. P. Clark, *Molecular Biology: Understanding the Genetic Revolution.* Oxford, UK: Elsevier, 2005.

[16] F. H. C. Crick, “On Protein Synthesis,” *Symp Soc Exp Biol.*, vol. 12, pp. 138–63, 1958.

[17] V. Vinson and E. Pennisi, “The allure of synthetic biology,” *Science*, vol. 333, no. 6047, pp. 1235–1319, 2011. [Online]. Available: http://www.sciencemag.org/journals

[18] D. Endy, “Foundations for engineering biology.” *Nature*, vol. 438, no. 7067, pp. 449–53, Nov. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16306983

[19] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, “Synthetic biology: new engineering rules for an emerging discipline.” *Molecular systems biology*, vol. 2, p. 2006.0028, Jan. 2006. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1681505\&tool=pmcentrez\&rendertype=abstract

[20] European Commission and N. H.-l. E. Group, “Synthetic Biology Applying Engineering to Biology,” European Comission, Brussels, Tech. Rep., 2005.

[21] T. M. Tumpey, C. F. Basler, P. V. Aguilar, H. Zeng, A. Solórzano, D. E. Swayne, N. J. Cox, J. M. Katz, J. K. Taubenberger, P. Palese, and A. García-Sastre, “Characterization of the reconstructed 1918 Spanish influenza pandemic virus.” *Science (New York, N.Y.)*, vol. 310, no. 5745, pp. 77–80, Oct. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16210530

[22] J. Kaiser, “Resurrected influenza virus yields secrets of deadly 1918 pandemic,” *Science Magazine*, vol. 310, 2005. [Online]. Available: http://www.sciencemag.org/content/310/5745/77.abstract

[23] C. J. Venter, “Gene synthesis technology: State of the Science National Science Advisory Board on Biosecurity (conference).” [Online]. Available: http://www.webconferences.com/nihnsabb/july_1_2005.html

[24] R. Weiss, S. Basu, S. Hooshangi, A. Kalmbach, D. Karig, R. Mehreja, and I. Netravali, "Genetic circuit building blocks for cellular computation, communications, and signal processing," *Natural Computing*, vol. 2, no. 1, pp. 47–84, 2003. [Online]. Available: http://www.springerlink.com/index/H885L73711912672.pdf

[25] J. W. Chin, "Modular approaches to expanding the functions of living matter," *Nature Chemical Biology*, vol. 2, no. 6, pp. 304–11, Jun. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16710339

[26] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators." *Nature*, vol. 403, no. 6767, pp. 335–8, Jan. 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10659856

[27] H. T. Baytekin and E. U. Akkaya, "A molecular NAND gate based on Watson-Crick base pairing." *Organic Letters*, vol. 2, no. 12, pp. 1725–1727, 2000.

[28] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology in Practice: Concepts, Implementation and Application.* Berlin, Germany: Wiley, 2005.

[29] D. B. Kell and J. D. Knowles, "The role of modeling in systems biology," in *System Modeling in Cell Biology: From Concepts to Nuts and Bolts*, Z. Szallansi, J. Stelling, and Vipul Periwal, Eds. Cambridge, MA: MIT Press, 2006.

[30] "Parts Registry." [Online]. Available: http://partsregistry.org/Main_Page

[31] "BioBricks Foundation." [Online]. Available: http://biobricks.org/

[32] P. Waage, C. M. Guldberg, and H. I. Abrash, "Studies concerning affinity," *Journal of Chemical Education*, vol. 63, no. 12, pp. 1044–1047, 1986.

[33] R. S. Goody and K. A. Johnson, "The original Michaelis constant: translation of the 1913 MichaelisâĂŞMenten paper," *Biochemistry*, vol. 50, no. 39, pp. 8264–8269, 2011.

[34] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, "Stochastic gene expression in a single cell." *Science (New York, N.Y.)*, vol. 297, no. 5584, pp. 1183–6, Aug. 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12183631

[35] A. Arkin, J. Ross, and H. H. Mcadams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected e-coli cells," *Genetics*, vol. 149, pp. 1633–1648, 1998.

[36] H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends in Genetics: TIG*, vol. 15, no. 2, pp. 65–9, Feb. 1999. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10098409

[37] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *Physical Chemistry*, vol. 105, pp. 1876–1889, 1999.

[38] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, pp. 403–434, 1976.

[39] D. McQuarry, “Stochastic approach to chemical kinetics,” *Journal of Applied Probability*, vol. 4, pp. 413–478, 1967.

[40] D. T. Gillespie and L. R. Petzold, “Numerical simulation for biochemical kinetics,” in *System Modeling in Cellular Biology*, Z. Szallasi, J. Stelling, and V. Periwal, Eds. London, England: MIT Press, 2006, ch. 16, pp. 331–353.

[41] T. Wilhelm, “The smallest chemical reaction system with bistability.” *BMC systems biology*, vol. 3, p. 90, Jan. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2749052\&tool=pmcentrez\&rendertype=abstract

[42] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY: Springer, 2002.

[43] H. Kuwahara, C. Madsen, I. Mura, C. Myers, A. Tejeda, and C. Winstead, “Efficient stochastic simulation to analyze targeted properties of biological systems,” in *Stochastic Control*, C. Myers, Ed. Sciyo, ch. 25. [Online]. Available: sciyo.com

[44] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar, “Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae,” *Bioinformatics*, vol. 18, no. 2, pp. 351–361, 2002.

[45] C. Reder, “Metabolic control theory: a structural approach.” *Journal of theoretical biology*, vol. 135, no. 2, pp. 175–201, Nov. 1988. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/3267767

[46] G. Strang, *Introduction to Linear Algebra*, 3rd ed. Wellesley, MA: Wellesley-Cambridge, 2009.

[47] J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler, “Mechanisms of noise-resistance in genetic oscillators.” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 9, pp. 5988–92, Apr. 2002. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=122889\&tool=pmcentrez\&rendertype=abstract

[48] H. Kuwahara, C. Myers, M. Samoilov, N. Barker, and A. Arkin, “Automated Abstraction Methodology for Genetic Regulatory Networks,” *Transactions on Computational Systems Biology*, no. IV, pp. 150–175, 2006.

[49] H. Kuwahara, C. J. Myers, and M. S. Samoilov, “Temperature control of fimbriation circuit switch in uropathogenic escherichia coli: quantitative analysis via automated model abstraction,” *PLoS Computational Biology*, vol. 6, no. 3, p. e1000723, Mar. 2010. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2845655\&tool=pmcentrez\&rendertype=abstract

Appendices

Appendix A

Chemical Reaction Network Models

A.1 Michaelis-Menten Enzymatic Reaction Network

Table A.1: Michaelis-Menten Reaction Network.

Reaction	Rate Constant
$E + S \xrightarrow{k_1} ES$	$k_1 = 0.005$
$ES \xrightarrow{k_2} E + S$	$k_2 = 0.1$
$ES \xrightarrow{k_3} E + P$	$k_3 = 1$

Table A.2: Reaction rate constants for the Michaelis-Menten model in Table A.1.

Species	Amount
E	50
S	100
ES	0
P	0

$$N = \begin{array}{cc} & \begin{array}{cccc} E & S & ES & P \end{array} \\ \begin{array}{c} R_1 \\ R_2 \\ R_3 \end{array} & \begin{pmatrix} -1 & -1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 \end{pmatrix} \end{array}$$

Fig. A.1: Stoichiometry matrix for the Michaelis-Menten model.

A.2 VKBL Circadian Rhythm Reaction Network Model

Table A.3: VKBL circadian rhythm model reaction network.

Reaction	Rate Constant
$DA \xrightarrow{k_1} DA + MA$	$k_1 = 50$
$DAp \xrightarrow{k_2} DAp + MA$	$k_2 = 500$
$DR \xrightarrow{k_3} DR + MR$	$k_3 = 0.01$
$DRp \xrightarrow{k_4} DRp + MR$	$k_4 = 50$
$MA \xrightarrow{k_5} A + MA$	$k_5 = 50$
$MR \xrightarrow{k_6} R + MR$	$k_6 = 5$
$A + DA \xrightarrow{k_7} DAp$	$k_7 = 1$
$A + R \xrightarrow{k_8} C$	$k_8 = 2$
$A + DR \xrightarrow{k_9} DRp$	$k_9 = 1$
$A \xrightarrow{k_{10}} \emptyset$	$k_{10} = 1$
$C \xrightarrow{k_{11}} R$	$k_{11} = 1$
$MA \xrightarrow{k_{12}} \emptyset$	$k_{12} = 10$
$MR \xrightarrow{k_{13}} \emptyset$	$k_{13} = 0.5$
$R \xrightarrow{k_{14}} \emptyset$	$k_{14} = 0.2$
$DAp \xrightarrow{k_{15}} A + DA$	$k_{15} = 50$
$DRp \xrightarrow{k_{16}} A + DR$	$k_{16} = 100$

Table A.4: Reaction rate constants for the VKBL model in Table A.3.

Species	Amount
A	0
C	0
DA	1
DAp	0
DR	1
DRp	0
MA	0
MR	0
R	0

$$N = \begin{array}{c|ccccccccc} & A & C & DA & DAp & DR & DRp & MA & MR & R \\ \hline R_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ R_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ R_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ R_5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ R_7 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ R_8 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ R_9 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ R_{10} & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_{11} & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ R_{12} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ R_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ R_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ R_{15} & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ R_{16} & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{array}$$

Fig. A.2: Stoichiometry matrix for the VKBL model.