

**Stochastic Transient Analysis of BioChemical Systems and Its
Application to the Design of Biochemical Digital Circuits**

**A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY**

Bin Cheng

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Master of Science in Electrical Engineering**

November, 2008

© Bin Cheng 2008
ALL RIGHTS RESERVED

Stochastic Transient Analysis of BioChemical Systems and Its Application to the Design of Biochemical Digital Circuits

by Bin Cheng

ABSTRACT

There are two main models for describing the reaction dynamics of biochemical systems: the deterministic continuous model and the stochastic discrete model. The former is formulated by a set of coupled ordinary differential equations (ODEs) and can be solved using some numerical methods such as Euler and Runge-Kutta. The latter, although has been proved to have a firmer physical basis than the deterministic continuous model and never approximate infinitesimal time increments, has no direct numerical method to solve, but adopts some Monte Carlo procedure to numerically simulate the system. To better understand the time behavior of the biochemical system, it is sometimes necessary and helpful to introduce various types of input signals to the simulation process. Such input signals are set to be functions of time rather than just static initial conditions. For deterministic continuous model, this can be analyzed through solving ODEs or even differential algebraic equations (DAEs) during the simulation, like what has been done in transient analysis of electrical circuits. However, such transient analysis method cannot be used in stochastic model, which sometimes is more accurate in describing the system than deterministic model, especially for cell-scale systems.

To better simulate and analyze the temporal behavior of biochemical systems, we incorporate a sufficient transient analysis into the stochastic model, define various time-dependent input signals and develop the corresponding algorithm for the simulation. First we apply the proposed transient analysis to a simple chemical model called Lotka and perform various simulations on it. Then we demonstrate how to use it to assist the design of logic gates in biochemical systems. Lastly it is applied to verify the design of a simplified inverter and a 3-stage ring oscillator in biochemical systems. Through the simulations, we demonstrate that the proposed method can yield new insights into the dynamics of biochemical systems; specifically it can be used to analyze and verify the design of biochemical digital circuits.

Acknowledgements

Firstly I would like to express my deepest thanks to my advisor, Marc Riedel, for providing the research direction and encouraging me to use my knowledge of electrical circuits into the study of biochemical systems. His precious expertise in digital circuits and biochemical systems plays a very important role of my research. With lots of insightful suggestions, he inspires me to develop the simulation method for biochemical systems proposed in this thesis.

I am very honored and appreciated to have had Professor Sachin Sapatnekar, Fellow of IEEE, in my thesis committee. Most of my advanced knowledge of circuits simulation is learned from his course. And his tremendous depth and breadth of knowledge in this field as well as his sense of humor make his course a very fruitful and happy learning experience.

I would also like to thank Professor Tian He, member of ACM and IEEE, who is from Computer Science but also a graduate faculty member in Electrical and Computer Engineering. I was fortunately agreed to attend his advanced course about wireless sensor network and had expanded my knowledge a lot in wireless network and embedded system. Many thanks to him for helping me review this thesis.

Lastly I appreciate all the help from the colleagues in our research group, especially those inspiring discussions in the group meetings.

Dedication

To my beloved daughter, Jinqi.

Contents

Abstract	i
Acknowledgements	ii
Dedication	iii
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Modeling and Analysis of Biochemical Systems	1
1.2 Synthesis and Design	2
1.3 Contribution	3
1.4 Thesis Outline	4
2 Background	6
2.1 Modeling Biochemical Systems Deterministically	6
2.1.1 Chemical Reaction Kinetics	6
2.1.2 Deterministic Continuous Modeling	7
2.2 Stochastic Discrete Modeling	8
2.3 Transient Analysis	9
3 Stochastic Transient Analysis of Biochemical Systems	11
3.1 Categorizing Input Signal For STA	11
3.2 STA algorithm	14

4	Stochastic Transient Analysis of Lotka Model	18
4.1	Lotka Model	18
4.2	Analyzing Lotka Model by STA	19
5	Verification of the Design of Biochemical Logic Gates	23
5.1	Biochemical Inverter	23
5.2	Some Other Logic Gates	27
6	A Simplified Inverter and A 3-stage Ring Oscillator	32
6.1	Design of A Simplified Inverter	32
6.2	Simulation of A 3-stage Ring Oscillator	35
7	Conclusion and Discussion	38
	Appendix A. Acronyms	39
A.1	Acronyms	39
	Appendix B. Reaction Parameter Tables	40
B.1	Reaction Parameter Tables	40

List of Tables

A.1	Acronyms	39
B.1	Species Initial Quantities in Lotka Model	40
B.2	Reaction Rates in Lotka Model	40
B.3	Species Initial Quantities of Weiss' Inverter	41
B.4	Reaction Rates of Weiss' Inverter	41
B.5	Reaction Rates of AND gate	42
B.6	Reaction Rates of Simplified Inverter	42
B.7	Reaction Rates of 3-stage Ring Oscillator	43

List of Figures

2.1	A simple RC circuit.	9
2.2	Transient analysis of the simple RC circuit.	10
3.1	Time segmenting for multi-trial STA.	16
4.1	Transient analysis of Lotka model with time from 0 to 20.	19
4.2	Transient analysis of Lotka model with system entering the state ($y_1=0, y_2=0$).	20
4.3	Transient analysis of Lotka model with system entering the state ($y_1=\infty, y_2=0$).	20
4.4	Transient analysis of Lotka model with forcing input for y_1	21
4.5	Transient analysis of Lotka model with pulse input for x	22
5.1	Stochastic transient analysis of the INVERTER.	26
5.2	Stochastic transient analysis of the AND gate.	28
5.3	Stochastic transient analysis of the NAND gate.	29
5.4	Stochastic transient analysis of the OR gate.	30
5.5	Stochastic transient analysis of the XOR gate.	30
5.6	Stochastic transient analysis of the XOR gate with lower frequency.	31
6.1	A simplified inverter.	34
6.2	A simplified inverter with different threshold.	34
6.3	A 3-stage ring oscillator.	35
6.4	A 3-stage biochemical ring oscillator.	36
6.5	A 3-stage biochemical ring oscillator with longer period and less amplitude variation.	37

Chapter 1

Introduction

1.1 Modeling and Analysis of Biochemical Systems

The traditional way to simulate the time evolution of biochemical systems is referred as continuous deterministic method. In this method, the system is modeled as a set of coupled ordinary differential equations (ODEs) or even differential algebraic equations (DAEs) ([1], [2]) based on the law of mass action. Although in most cases, continuous deterministic simulation is able to accurately capture the time behavior of the system, it loses the accuracy when it comes to the biochemical systems formed by living cells. This is not surprising because the physical nature of a chemical reaction is actually some kind of collision among molecules, which is stochastic and discrete rather than deterministic and continuous. In a cell-scale system, such inherent stochasticity of biochemical systems appears more strongly.

To catch the random manner of biochemical reactions, Gillespie proposes a stochastic simulation algorithm (SSA) in [3] and demonstrates that his method has a firmer physical basis than the continuous deterministic method and can bring more accurate simulation results. The SSA works perfectly for small systems in which the quantities of molecules are on the order of no more than thousands. However, the high computational cost of SSA simulation prohibits it from being used for large systems. In view of such limitation of SSA, an improved algorithm is proposed in [4]. And to achieve even better performance in terms of speed, several other methods have been proposed focusing on adopting approximations during the simulation process as in [5], [6], [7], [8], [9].

So far, all the relevant research has been focusing on how to speed up the stochastic simulation and how to use it to analyze and synthesize biochemical systems ([10], [11]). A typical scenario is, given the initial state of the system, namely the quantities of each species, we simulate the system using stochastic simulation to get the trajectory of each species along the time dimension or the probability of some state at certain time point. Here the system and its initial state are always assumed static, i.e. no external drive for any species exists. But what if we want to study the time behavior of the system when there exists some other mechanism which also affects the quantities of some species. For instance, we may need to study how to make some system stable by injecting certain species into the system periodically or when some event fires (e.g. the quantity of some species is below certain threshold). Under such circumstance, the system cannot be simulated using the existing stochastic simulation methods without any change. In this thesis, a renovated approach is presented to resolve such issue.

1.2 Synthesis and Design

Similar to the systems in other domain, such as computer science or electrical engineering, the synthesis and design technologies play the most important role in biochemical systems as they act as the interfaces that make it possible to apply all the relevant researches into the real practical life and benefit the society and the whole human beings. Without them, no matter how marvelous and perfect the modeling and analyzing approaches are, the domain can only stay on peer theoretical level, which is almost unacceptable for an engineering field. However, accurate and precise modeling and analyzing approaches are also critical to the process of synthesis and design. Nowadays, most systems are so complicated that it is almost impossible to succeed in building real systems simply based on the design without any simulation and verification.

In biochemical systems, design often falls into two categories [12]: one is use natural biological components to design a new system with specific function; the other is build a system by hooking together a series of chemical reactions which do not exist in nature. Compared with the former, the latter approach is easier to simulate and verify. The reason is because both deterministic and stochastic simulations can only get quantitative results if the system is modeled mathematically. The latter can be accurately modeled by chemical reaction kinetics while there is no direct mathematical modeling way for the former. Therefore, to simulate the former, models in chemical reactions need to be built to approximate the function of natural components.

Such models are either coming out from experiments and personal experience, or from certain systematic synthesizing method, such as [10] presents a stochastic synthetic method which can build a chemical reactions model implementing a probabilistic response of the lambda bacteriophage. In [13], authors design a biochemical oscillator through a mathematical model of transcription regulation. They all use either deterministic continuous simulation or stochastic discrete simulation or both to assist and verify their synthesis and design.

A new stochastic method is presented in this thesis for simulating small-scale biochemical systems, especially for verifying the design of biochemical digital circuits.

1.3 Contribution

In this thesis, we propose a systematic approach called stochastic transient analysis (STA) to solve the problem aforementioned. Strictly speaking, stochastic simulation is already a transient method as it simulates the time-dependent behavior of the system. In electrical circuits, transient analysis is a time sweep using numerical methods to solve the differential algebraic equations of the circuits with the operating points solved by DC analysis as the initial conditions. Similar to the transient analysis in electrical circuit simulation, for a biochemical system, various input signals can be defined and the system can be solved through ODEs or DAEs using the continuous deterministic method. Actually it is reported that electrical simulation tool SPICE [14] has been used to model various biosynthesis reactions [15], [16], [17]. And in [18], authors even present a systematical approach for modeling biochemical systems using SPICE. However, for cell-scale biochemical systems, stochastic simulation needs to be adopted to get more accuracy and the input signal form is more complex than electrical circuits. For example, an accumulative input signal can be injected into biochemical systems while it cannot for electrical circuits.

From the discussion above, we see that although the stochastic simulation itself already implies transient analysis, all the existing stochastic methods cannot deal with the situations when there exists some external driving force which injects or forces the quantity of some species with a function of time. So in our proposed stochastic transient analysis, we first systematically categorize various input signal forms based on the feature of biochemical system and then modify the stochastic simulation algorithm accordingly.

In STA, input signals are designed to satisfy different application scenarios. The input functions are categorized into pulse, piecewise linear, sinusoidal and some other free-form functions. And the signal imposed mode is either forced or injected. Further, event-driven signals are also incorporated. The algorithm describes how to deal with different input signals during the stochastic simulation. A simple model called Lotka is presented to demonstrate the capability of the proposed STA.

Specifically for the design of digital circuits in biochemical systems, STA can provide the transient analysis similar to that in electrical circuit simulation. Actually there already exists a system called BioSPICE [19], which is an integrated environment for simulating biochemical systems like SPICE for simulating electrical circuits. However existing approaches for analyzing biochemical circuits are using continuous deterministic approach [20], [21], which might not be accurate enough for cell-scale systems. In fact, a 1-bit ALU built on DNA transcription logic has even been proposed in [22]. But the discussion there seems to be quite brief and the detailed reactions and simulation methods are not given out. In this paper, we use STA to analyze the time behavior of the inverter proposed in [20], where it is originally analyzed through continuous way. And STA is also used to verify the design of some other logic gates in biochemical systems, such as AND, OR, NAND and XOR gates. Lastly we design a simplified inverter and a 3-stage ring oscillator. STA is applied to verify our design.

1.4 Thesis Outline

- Chapter 2 briefly presents the background knowledge about stochastic simulation algorithm and circuit transient analysis.
- In Chapter 3, the proposed stochastic transient analysis is described.
- In Chapter 4, stochastic transient analysis is applied to simulate and analyze a simple biochemical model - Lotka.
- In Chapter 5, stochastic transient analysis is used to verify the design of logic gates.
- Chapter 6 presents the design of a simplified inverter and a 3-stage ring oscillator as well as their simulation results of stochastic transient analysis.

- Chapter 7 presents a conclusion of the thesis and the discussion of the further investigation of STA in the future.

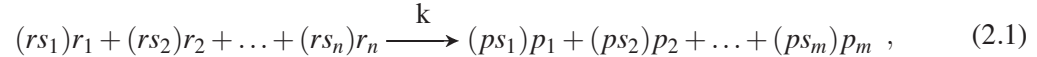
Chapter 2

Background

2.1 Modeling Biochemical Systems Deterministically

2.1.1 Chemical Reaction Kinetics

We study biochemical systems by chemical reactions. Suppose we write a chemical reaction as follows:



where $r_i (i = 1, 2, \dots, n)$ represents a molecule in the system and is called reactant that is consumed in the chemical reaction. $p_i (i = 1, 2, \dots, m)$ also represents a molecule but is produced in the chemical reaction, thus is called product. $rs_i (i = 1, 2, \dots, n)$ and $ps_i (i = 1, 2, \dots, m)$ appear in front of reactants and products and are called stoichiometries, which represent the quantities of reactants and products in a chemical reaction. k is the rate constant of the reaction, which represents the inherent speed feature of the reaction. To analyze biochemical systems quantitatively, we must apply the law of mass action to our analysis.

The law of mass action was observed decades ago and the speed of chemical reaction can be determined by it. According to the law of mass action, the speed of a chemical reaction is proportional to the product of the concentrations of all the reactants. If we use R_i to represent the concentration of reactant r_i in reaction (2.1), then the time-rate of the reaction, V , can be

computed by:

$$V = k \prod_{i=1}^n R_i^{r_{S_i}} . \quad (2.2)$$

2.1.2 Deterministic Continuous Modeling

Suppose we have a simple biochemical system, in which there are only three species denoted by s_1, s_2, s_3 and their concentrations are represented by S_1, S_2, S_3 respectively. And it has the following chemical reaction set:



From the reaction set (2.3), we can see that two molecules of species s_1 are consumed in the first reaction and two are produced again in the second reaction. But this doesn't mean that the concentration of s_1 remains constant and doesn't vary with the time, because the two reactions have different reacting rates. According to equation (2.2), we get the time-rates of two reactions as follows:

$$\begin{array}{l} V_1 = k_1 S_1^2 S_2 \\ V_2 = k_2 S_3 \end{array} . \quad (2.4)$$

So the variation of S_1 over time can be specified by the following differential equation:

$$\frac{dS_1}{dt} = -2k_1 S_1^2 S_2 + 2k_2 S_3 . \quad (2.5)$$

Similarly we can get the differential equations for S_1 and S_2 as follows:

$$\frac{dS_2}{dt} = -k_1 S_1^2 S_2 , \quad (2.6)$$

$$\frac{dS_3}{dt} = 2k_1 S_1^2 S_2 - k_2 S_3 . \quad (2.7)$$

However, when the system is relatively small, for example a cell, then quantities of molecules

are usually no more than thousands or even less. With such small number of molecules, the system may not be well stirred and the species may not be distributed evenly among the system environment. Under such circumstance, a uniform concentration can hardly describe the distribution of a species in the system. Moreover, continuous differential equations cannot accurately approximate the dynamics of the system due to small quantities of molecules.

To overcome such difficult situation, a new approach, stochastic discrete modeling, is proposed. And it is described briefly in the next section.

2.2 Stochastic Discrete Modeling

In [3], the author proposes a stochastic simulation algorithm (SSA) for biochemical systems, which has been proven to have a firmer physical basis than the deterministic formulation. Further, it is an exact approach in the sense that it never approximates infinitesimal time increment dt by finite time step δt like most numerical methods do for solving deterministic ODEs.

The main idea of SSA is as follows: in a biochemical system which contains a spatially homogeneous mixture of the molecules of some chemical species, the sequence of reactions that fire behaves as a Markov process, i.e. the probability of next reaction depends only on the present state of the system, not on the past ones. More precisely, the probability that a reaction is next to fire is proportional to both the number of distinct molecular reactant combinations available in the current system state for all the reactants in the reaction and its rate which depends on the system volume, temperature and other physical parameters. As a result, if we define a_i as in (2.8), we have the formulation in (2.9) to represent the probability of i th reaction that fires next.

$$a_i \equiv h_i c_i, \quad (2.8)$$

where h_i is the number of distinct molecular reactant combinations available in the current system state for all the reactants in the i th reaction and c_i is the stochastic reaction constant of the i th reaction.

$$P_i = a_i / a_0, \quad (2.9)$$

where $a_0 = \sum a_i$.

To simulate the time evolution behavior of the system, SSA also gives the formulation of the time step when the next reaction fires in (2.10).

$$\tau = (1/a_0)\ln(1/r), \quad (2.10)$$

where r is a random number generated by a unit-interval uniform random number generator.

2.3 Transient Analysis

Transient analysis is the way to simulate the time evolution of the system. Specifically for electrical circuits, transient analysis is used to analyze the circuit by solving circuit equations, usually ODE or DAE, using numerical methods, such as Euler, Trapezoidal, Gear etc. For example, Fig. 2.1 displays a simple RC circuit. According to MNA (Modified Nodal Analysis), we can get a differential equation for the voltage $v(t)$. Solving the equation numerically, we can obtain the time response curve of $v(t)$ to the pulse input signal as depicted in Fig. 2.2.

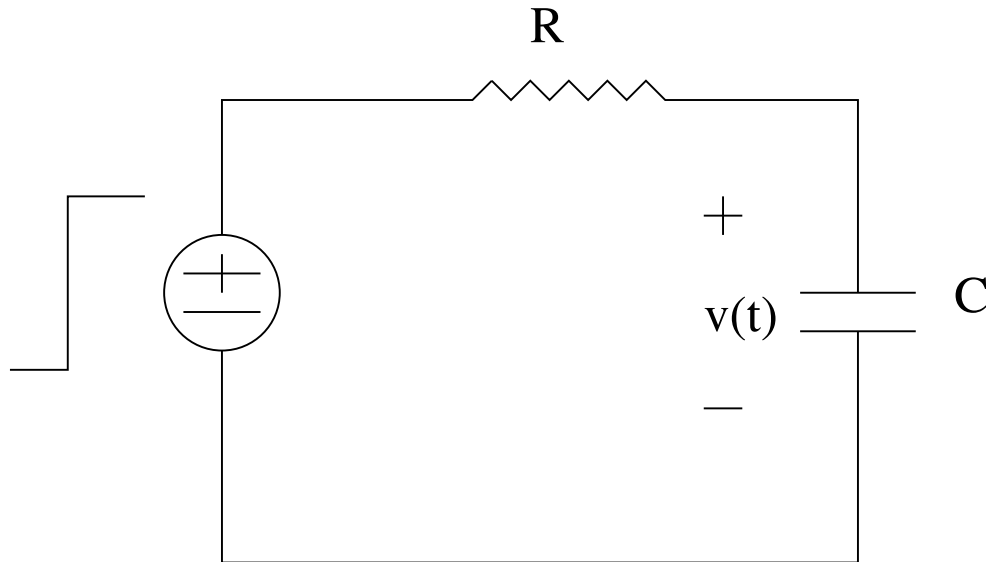


Figure 2.1: A simple RC circuit.

Similarly, for the deterministic continuous model of biochemical systems, where the system is modeled by differential equations, the same approach can be applied to get the time response

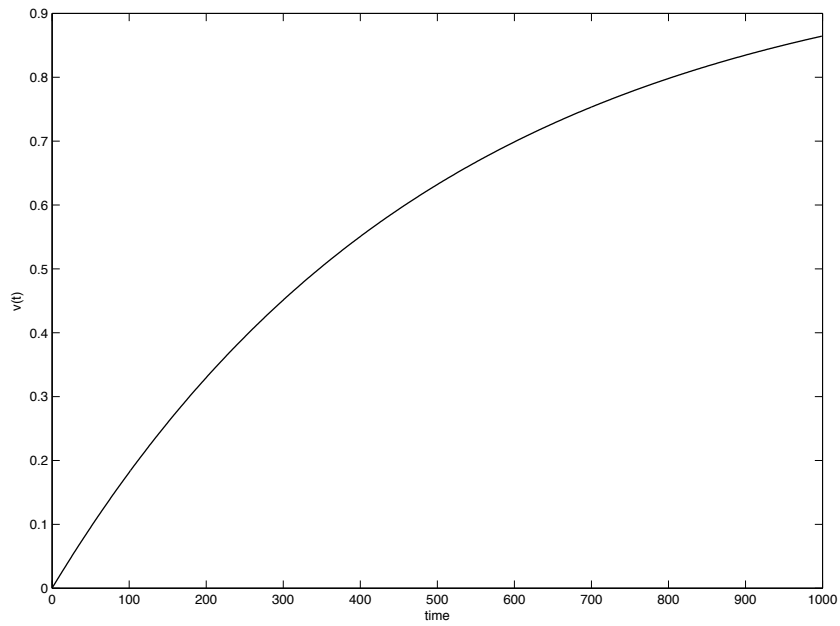


Figure 2.2: Transient analysis of the simple RC circuit.

of any species in the system. However, for stochastic discrete model, numerical method cannot be applied to solve the system, because there are no ODE or DAE equations in the model. This thesis is focusing on how to perform stochastic discrete transient analysis to biochemical systems.

Chapter 3

Stochastic Transient Analysis of Biochemical Systems

The power of transient analysis resides in the fact that the time behavior of the system can be simulated under different input contexts. The static initial state of the system in the existing stochastic methods is a special case of various forms of input signal. A powerful sufficient transient analysis should be able to deal with many kinds of input signals to accommodate different analytical scenarios. In this section, first the input signal is categorized systematically for stochastic transient analysis based on the characteristics of biochemical system, then the proposed STA is described in detail.

3.1 Categorizing Input Signal For STA

Transient analysis has been used to simulate electrical circuits for decades and has been developed quite maturely and sufficiently. Although the deterministic continuous approach as adopted in the electrical transient analysis is not discussed for biochemical systems in this paper, the way that the electrical transient analysis organizes input signal can be borrowed for STA.

In electrical circuit simulation system, input signal for transient analysis is categorized in three main forms: pulse, piecewise linear (PWL) and sinusoidal as described as follows respectively: $PULSE(v1\ v2\ td\ tr\ tf\ pw\ per)$, where $v1$ is initial value, $v2$ is pulsed value, td is delay time, tr is rise time, tf is fall time, pw is pulse width and per is period; $PWL(t1\ v1\ t2\ v2\ \dots)$,

where v_1 is the input value at time t_1 and v_2 at t_2 etc.; $SIN(v_0 \text{ va } freq \text{ td } theta \text{ ph})$, where v_0 is offset, va is amplitude, $freq$ is frequency, td is delay, $theta$ is damping factor and ph is phase. Apparently pulse and PWL can be adopted in STA without any change because they are able to capture most input signal scenarios with or without a period respectively. As to the sinusoidal signal, since it is almost indispensable for analyzing electrical circuits, it can also be expected to be an important signal form for analyzing biochemical circuits. As a result, all the input signal forms for the transient analysis of electrical circuits are adopted in STA. Different from electrical circuits, biochemical systems might be injected or forced into with any linear or non-linear signals for different application or research scenarios. For example, the quantity of some species may be injected into a biochemical system by a linear function of time. Therefore, besides all the signal forms adopted from electrical circuits, an open function interface for input signals should be incorporated into STA to satisfy various scenarios.

Besides the need of defining input signals, the way of imposing them into the system should also be specified. In general, signals can be inputted into the system through two ways: forced or injected. In the stochastic simulation, the former means the quantity of some species is always determined by some external drive rather than any internal reactions. The latter, on the other hand, deals with the situation when the external drive only adds certain number of some species occasionally rather than dominates the whole situation. In other words, at certain time point, the quantity of that species is determined by both the internal reactions and the external drive. As a result, in the STA algorithm, a forced input signal can be sampled at certain time point according to the input signal form, while an injected input signal cannot. To explain this conclusion, we can use a very simple example. Suppose the input signal is defined as follows: $x \text{ 0 } PWL(0 \text{ 10 } 100 \text{ 10})$, where x is the species, the first 0 is the initial quantity of x , PWL means it is a piecewise linear signal, the following sequence of numbers 0 10 100 10 means from time point 0 to time point 100, the signal strength is always 10.

If the signal is forced, as we have pointed out, at any time point between 0 and 100, the quantity of species x can be sampled, which here is always 10. However, if it is injected, even at the time point 0, we enter into a dilemma about how to sample the input signal. Even if we assume the injection happens in no time, we can have 10 of species x at time 0, then how do we decide the sampling at the next time point. Suppose the first reaction in which species x gets involved with fires at time point t and 2 of species x is consumed during the reaction, then only $10 - 2 = 8$ of species x is left after that reaction. However, we face a difficulty to decide

how many species x has been injected again so far. Is it 10 again? It could be, but without any reasonable sense in terms of injection if we do not define the frequency of the injection. If we always assume the injection happens with no time and is always enforced, namely the injection is continuous and its frequency is infinity, then there is infinite number of species x already. Through the dilemma we encounter in the example, it can be clearly seen that either the injected signal is defined discretely by time for each injection or is defined by a function of the total injected quantities to the time. The latter is adopted in our algorithm as it is much simpler to deal with than the former. The subtraction of the quantity at the previous time point from the one at the current time point would be the injected quantity between the two time points.

$$\begin{aligned}
 &< SPECIES > < INITIAL > \\
 &[< FUNCTION > < FORCED | INJECTED >] [EVENT] \\
 &Comments : \\
 &SPECIES \quad \quad \quad - - \textit{species name} \\
 &INITIAL \quad \quad \quad - - \textit{initial quantity} \\
 &FUNCTION \quad \quad \quad - - \textit{input function definition} \\
 &FORCED|INJECTED \quad - - \textit{input imposed type} \\
 &EVENT \quad \quad \quad - - \textit{event definition}
 \end{aligned} \tag{3.1}$$

Another useful consideration is to incorporate event-driven input signal, which means when some event happens, the specified input signal starts. This is useful when we want to deal with some special situations. For example, when the quantity of some species reaches 0, some reactions cannot fire again. Then by defining an event-driven input signal, some number of that species can be injected into the system when it becomes 0. This way we can keep some reactions happening and study how the system behaves under such injection mechanism. Another example is, under some circumstance the quantity of some species might be increasing so dramatically that even in a very tiny time step it can be reaching a very large number. If this happens, the simulation program might get stalled or even crashed and no simulation result can be drawn to give an idea about what happens in the system. In view of this, an event-driven input signal can be defined to force the quantity of that species to some certain value representing infinity when it goes beyond that value.

The format of the input signal for STA is summarized in (3.1), where $\langle \rangle$ means the contents are mandatory while $[]$ means optional. Detailed format is not presented, which can be different

in terms of implementations.

It should be noted that, besides the three standard signal functions, the definition and evaluation of the free-form function can be different in terms of implementation. To standardize it, some model formatting technology can be employed, such as SBML. Another thing should be noticed that there may be more than one input signal defined for one species. This happens when some event-driven input signal is added to the species besides some basic input signal.

3.2 STA algorithm

The stochastic transient analysis is based on Gillespie's direct SSA proposed in [3]. It can also be incorporated into the other improved stochastic simulation algorithms, such as [4]. In this paper, the STA algorithm is described using the direct SSA for simplicity. The incorporation of it into the other stochastic algorithms can be implemented similarly.

Suppose we have a biochemical system, in which there are n different species interacting through m different reactions. The species types are denoted by $x_i (i = 1, \dots, n)$ and their corresponding quantities are $X_i (i = 1, \dots, n)$. The reactions are denoted by $R_i (i = 1, \dots, m)$ and the corresponding stochastic reaction constants are $c_i (i = 1, \dots, m)$. The system state at time point t is denoted by $S_t = (X_1(t), \dots, X_n(t))$. At the beginning of the simulation, the system state is $S_0 = (X_1(0), \dots, X_n(0))$, in which the quantity of each species is their initial value. To continue simulating the time-evolution of the system, the next reaction and the next time point at which the next reaction fires must be decided. According to Gillespie's direct SSA, they are determined by (3.2) and (3.3) respectively.

$$\sum_{v=1}^{\mu-1} a_v < r_1 a_0 \leq \sum_{v=1}^{\mu} a_v, \quad (3.2)$$

$$\tau = (1/a_0) \ln(1/r_2), \quad (3.3)$$

where μ is the next reaction that fires, τ is the time elapsing from now until the next reaction fires, r_1 and r_2 are two random numbers generated by a unit-interval uniform random number generator. a_0 is defined by (3.4), and a_v is defined by (3.5).

$$a_0 \equiv \sum_{v=1}^m a_v, \quad (3.4)$$

$$a_v \equiv h_v c_v (v = 1, \dots, m), \quad (3.5)$$

where

$$h_v \equiv \text{number of distinct } R_v \text{ molecular reactant} \\ \text{combinations available in the state } (X_1, \dots, X_n)$$

and c_v is the stochastic reaction constant for reaction R_v .

In STA, due to various types of input signals, the state of the system must be updated according to both the previous reaction and the evaluation of input signal at current time before the next reaction and the time point of the next reaction are calculated. From (3.1), we know that an input signal can consist of four parts: initial value, input function, imposed type, and event. If the input signal of a species only has the first part, i.e. initial value, then the update only considers the effect of the previous reaction, as described in original SSA. If the second and third parts are attached to the signal, then the way of update will depend on the imposed type of the signal. For the forced signal, the signal value is sampled from the function defined in second part at the current time point and the effect of the previous reaction takes no effect. While for the injected signal, since the injection is defined as a function of total injected quantity to time, so it must be calculated by subtracting the function value at the time point of previous reaction which has considered the injection effect from the function value at the current time point. If we use t_p representing the time point of previous reaction which has considered the injection effect and t_c representing the current time point and the evaluation of input function is f_p at time t_p and f_c at time t_c , then the number of species injected into the system since previous reaction which has considered the injection effect is represented by (3.6).

$$\text{injected number} = \text{floor}(f_c) - \text{floor}(f_p). \quad (3.6)$$

In (3.6), function $\text{floor}(x)$ returns the largest integral value that is not greater than x . If the injected number is greater than 0, then the current reaction will consider injection effect, which means for the next reaction, the previous reaction which has considered the injection effect will become the current reaction. Otherwise it will remain the old one.

Once the injected number of an input signal for the next reaction is calculated, then simply

adding the injected number to that species as well as considering the effect of previous reaction will get the updated quantity of the species.

If the fourth part *event* appears in the definition of the input signal, then a forced or injected signal will be defined inside the event and applied only when the event happens. The input function of the signal will be evaluated with the time when the event happens as the start time 0.

Due to the stochasticity of biochemical systems, sometimes multiple trials of simulation must be performed to get more accurate results. Under such situation, simply recording system state at each time point for all the simulation trials is not a good solution for transient analysis. Because with trails increasing, the data need to be recorded are also increasing proportionally. As a result, the program may slow down a lot or even run out of memory when the the number of trails is beyond certain level. To overcome this issue, we propose the following data structure which only consumes constant of memory no matter how many trials are performed.

Suppose total time of transient simulation performed is T , and the total time steps drawn in the final result curve is set to N . Then we separate T into N time segments equally, which are $[(i-1)T/N, iT/N](i = 1, \dots, N)$. For each time segment, a general state is attached, say, $S^{(i)}(i = 1, \dots, N)$. Since in stochastic simulation the sequence of fired reactions and corresponding time points vary with trails, the time points and corresponding states must be considered separately for each trial. If the current time point falls in i th time segment, then the corresponding state is added up into the i th general state $S^{(i)}$. At the end of simulation, each general state is averaged being divided by total number of simulation steps in the corresponding time segment. The final transient results can be plotted by using N general states at N time points. The data structure is shown in Fig. 3.1.

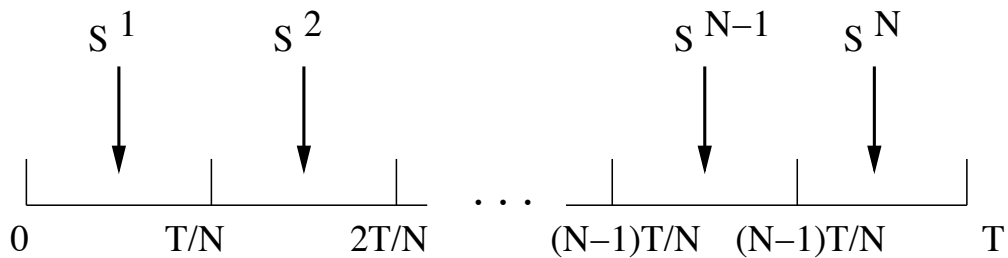


Figure 3.1: Time segmenting for multi-trial STA.

It is worth being noted that since the judgment of which time segment the current time point falls in happens at each simulation step, how to get the time segment index rapidly really matters in the simulation performance. Another thing should be noticed is that when there is no probability for some reaction to fire, namely the quantity of some reactant is not enough for the reaction to fire one time, it does not necessarily mean that the reaction will not be fired any more. Because the input signal of that deficient reactant may be forced or injected into the system again. All these are just some programming skills and they are not discussed in detail here.

To summarize, the pseudo codes for STA algorithm are listed as follows, which only describe the main logic while omit the initialization and some details.

```

for i=1 to trials
  time = 0
  while time <= T
    state = sampleInput(time, state)
    segment_no = getTimeSegment(time)
    SegState[segment_no] += state
    SegCount[segment_no] ++
    next_reaction = getSSANextReaction(state)
    tau = getSSATau(state)
    state = updateState(state)
    time += tau
  end while
end for

for i=1 to time_segments
  SegState[i] /= SegCount[i]
end for

```

Chapter 4

Stochastic Transient Analysis of Lotka Model

In this section, we first apply our transient analysis to a simple biochemical model, Lotka. Through the simulation results, we demonstrate that how our approach can help bring more insights into the dynamics of the system.

4.1 Lotka Model

In [3], the author has performed several different stochastic analyses to the Lotka model, which is described by a set of coupled reactions in (4.1). However all of the analyses for Lotka model in [3] are performed by using different parameters or adding some extra reactions. Some conclusions are even made only through logical deductions. For instance, the author claims that "no matter what the state of the system is initially, it will eventually wind up in either the state $(Y_1=0, Y_2=0)$ or the state $(Y_1=\infty, Y_2=0)$." However no simulation results are provided to support such claim. In the following simulation process, we will show how to perform relevant transient analysis to support such claim and how we are going to do more various transient

analyses to help us understand this model better.

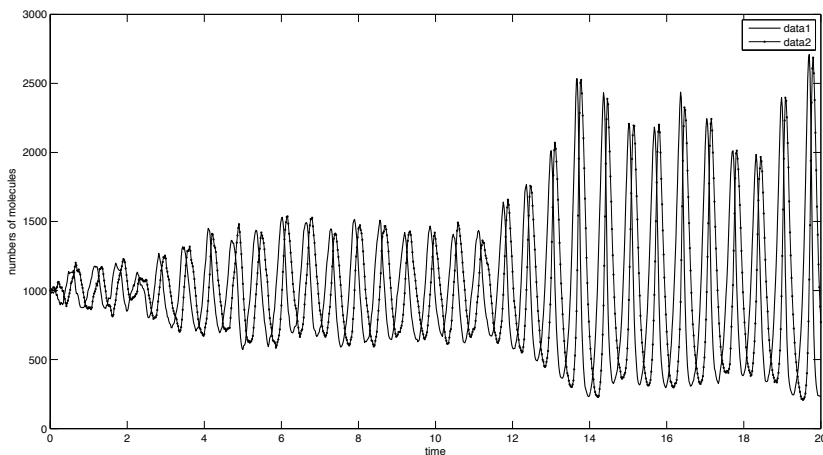
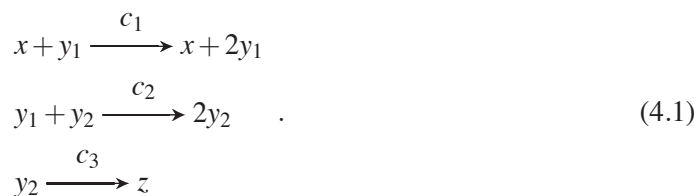


Figure 4.1: Transient analysis of Lotka model with time from 0 to 20.

4.2 Analyzing Lotka Model by STA

First let's perform the simulation to the model as done by the author of [3]. The result is shown in Fig. 4.1. Table B.1 in Appendix B describes the initial quantity of all species in the model, while the reaction rates are listed in Table B.2 in Appendix B.

Then we try to figure out what the simulation data would be for the situation when the system finally enters the state ($y_1=0, y_2=0$). To achieve this goal, we can simply prolong the simulation time. The result is shown in Fig. 4.2.

Now, to simulate how the system gets into the state ($y_1=\infty, y_2=0$), we cannot just prolong the simulation time or change the random number series. Because when y_1 was approaching to infinity, the time step for each fired reaction would be approaching infinitesimal, which means

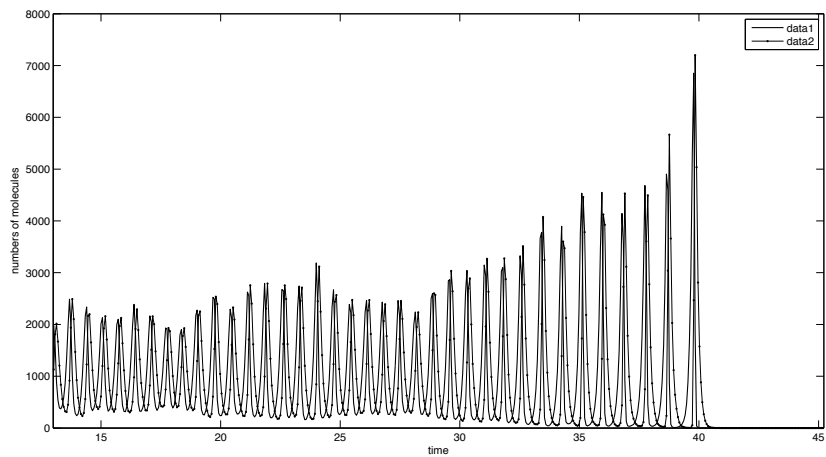


Figure 4.2: Transient analysis of Lotka model with system entering the state ($y_1=0$, $y_2=0$).

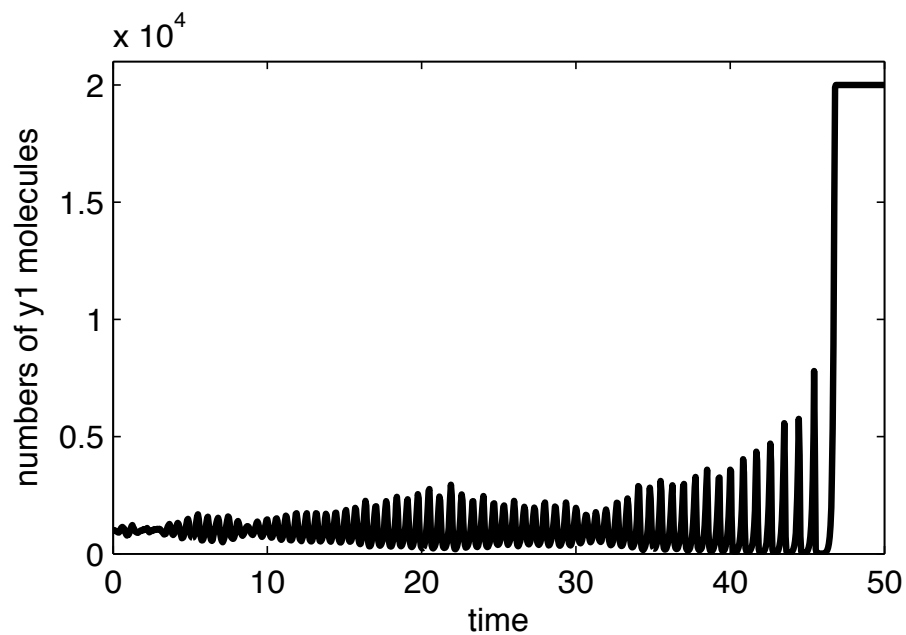


Figure 4.3: Transient analysis of Lotka model with system entering the state ($y_1=\infty$, $y_2=0$).

you would get nearly infinite reactions fired per time unit. The simulation data will simply blow up and crash your program. However in our transient analysis method, we can work around this

issue by defining event. If we constrain the output of y_1 to be 20,000 (approximating infinity) through event, i.e. forcing it to be 20,000 if beyond, we will get the waveform as in Fig. 4.3.

Suppose there is some mechanism from external which makes y_1 stay at 1000 constantly. Fig. 4.4 illustrates the simulation results with this forcing input for y_1 . Interestingly we observe that y_2 still finally gets to zero, even though it costs much more time.

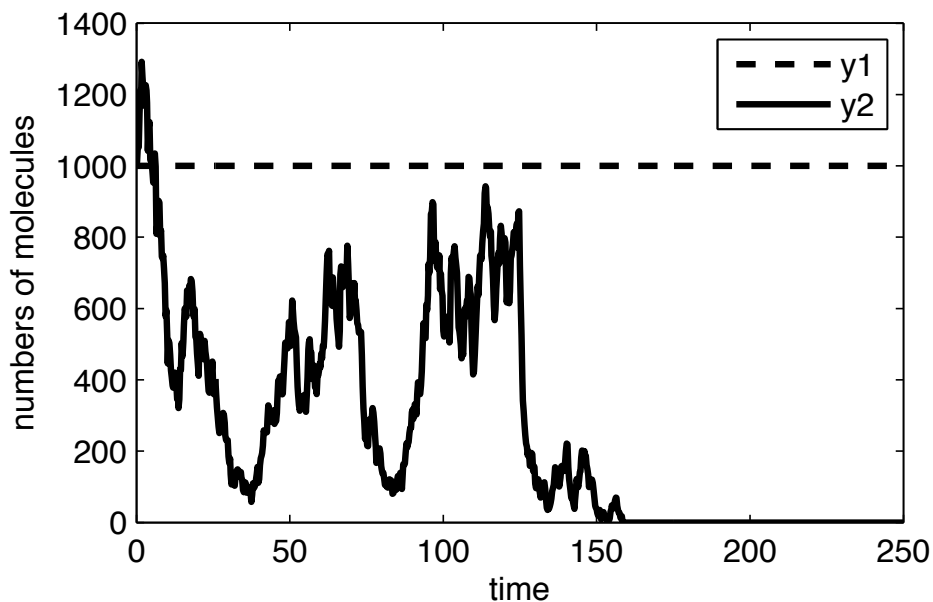


Figure 4.4: Transient analysis of Lotka model with forcing input for y_1 .

According to the deterministic steady-state analysis, the fluctuating center of y_2 is proportional to x , while y_1 's does not change with x . This can be verified through the simulation results depicted in Fig. 4.5, where x is set to be a pulse function of time: *PULSE*(10 30 50 0 0 50 100), as displayed in the top graph of Fig. 4.5. Here threshold events are defined: once y_1 or y_2 reaches zero, more of corresponding species is injected to keep the fluctuations going.

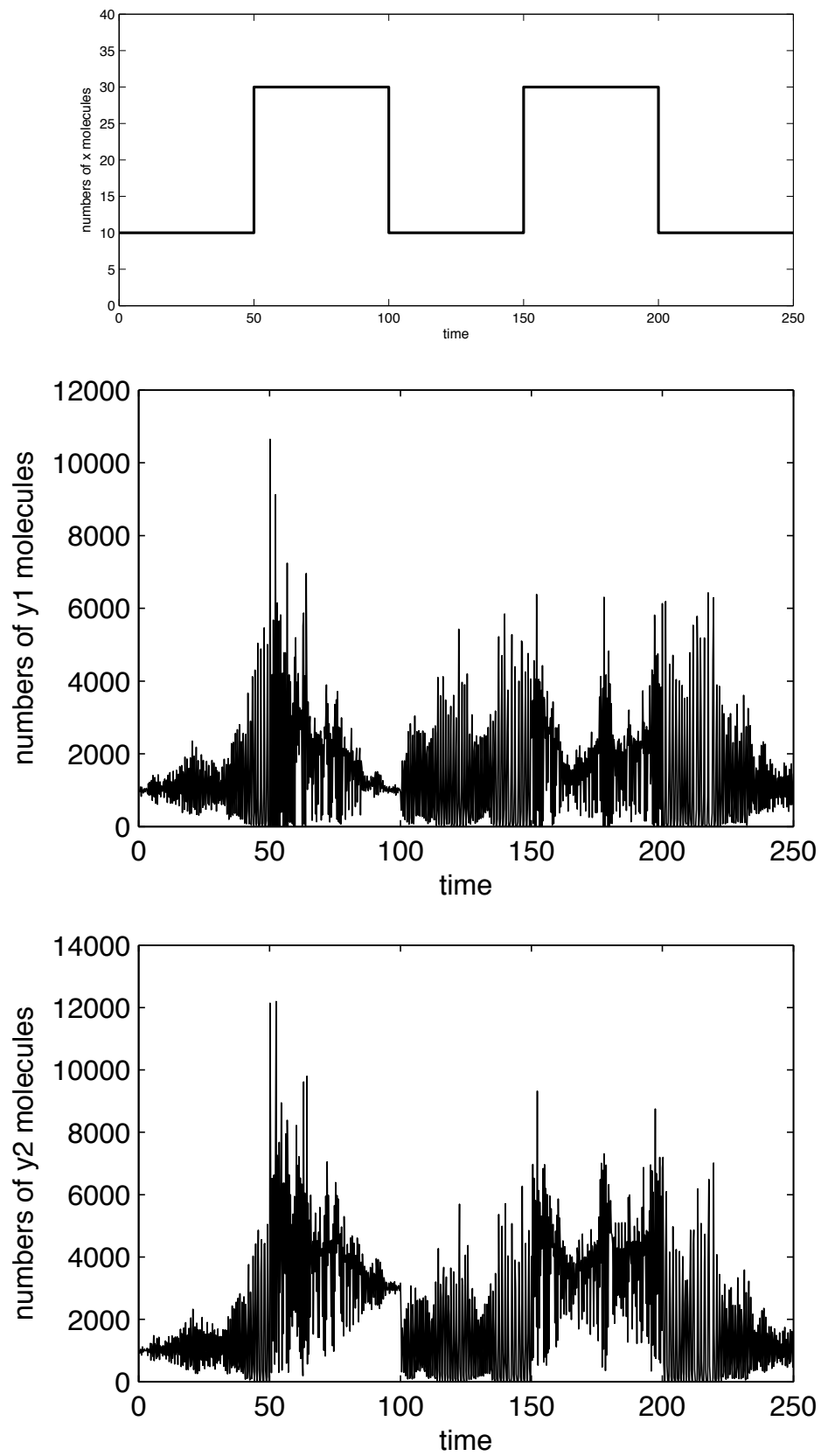


Figure 4.5: Transient analysis of Lotka model with pulse input for x .

Chapter 5

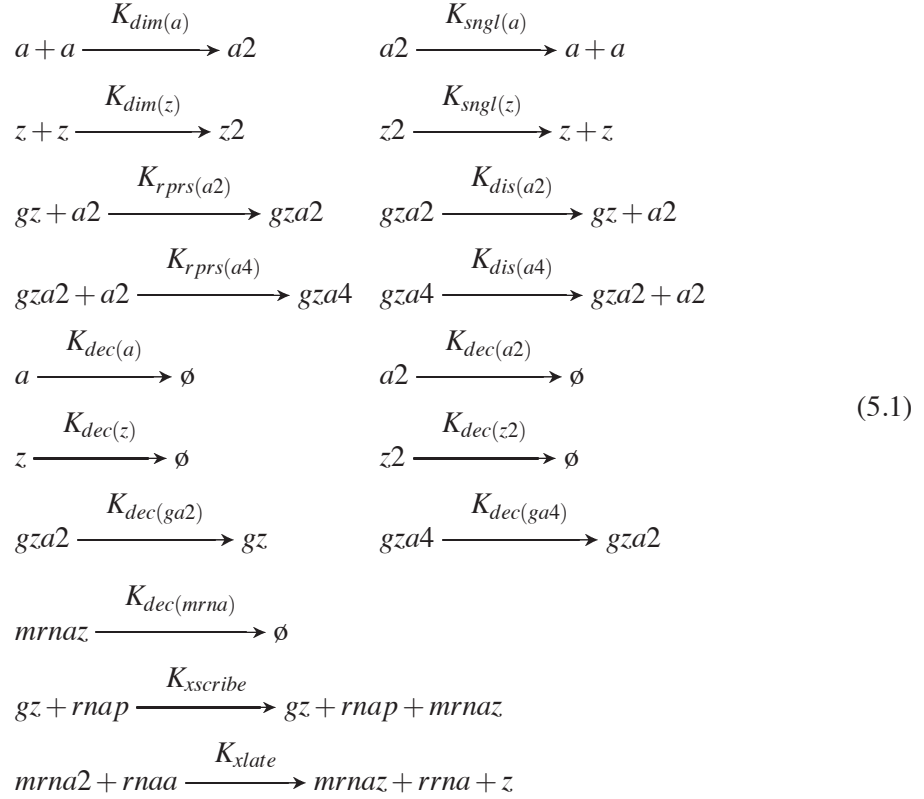
Verification of the Design of Biochemical Logic Gates

In this section, we apply STA to the logic gates design in biochemical system. First we employ an inverter model proposed in [20] and use the stochastic way to simulate the transient behavior of it. Then based on the inverter model and a simple AND model we propose, we create models for some other logic gates such as NAND, OR and XOR. We simulate all the models using STA to verify the correctness of our design.

5.1 Biochemical Inverter

In gene regulation systems, some features exhibit strong logic gates behavior. For instance, RNA polymerase will stop transcribing the gene if there exists a repressor protein which binds to the operator of the gene's promoter. If you consider the concentration of the gene and the concentration of the repressor as two signals, then the relationship between them is like an inverter. In [20], authors present a set of coupled chemical reactions to model such characteristics. The chemical reaction model is described in (5.1). Extra reaction is needed to maintain the quantity level of input species a , which is not listed here. And the initial species quantities and reaction

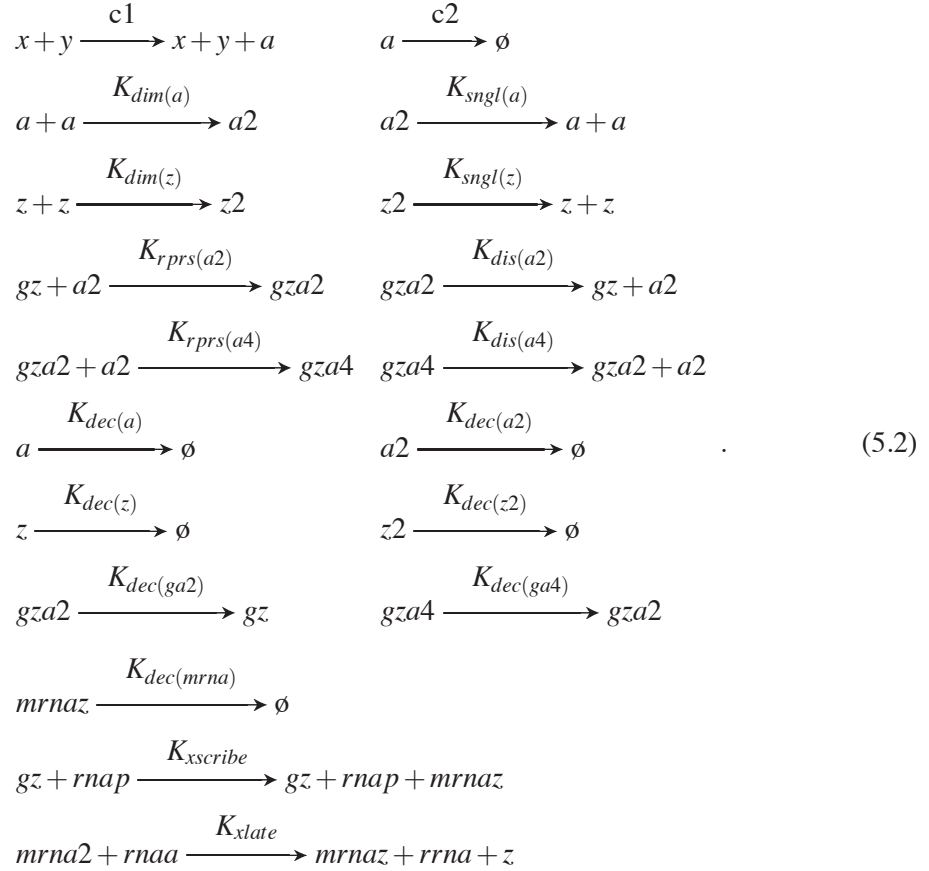
rates are listed in Table B.3 and Table B.4 respectively in Appendix B.



In [20], authors don't mention if they analyze the model using deterministic or stochastic approach. But in a late paper [21], which has the same first author as [20], authors claim that they are using deterministic way. Although they argue that in some sense deterministic model is equivalent to stochastic one, we will be using our stochastic transient analysis to analyze the inverter they have proposed and a little more logic gates we design based on their inverter for biochemical systems. By using stochastic analysis, we think it will bring more accurate results especially for cell-scale systems as stochastic model has a firmer physical basis than deterministic model.

To analyze the inverter using STA, we first provide logic '0' at the input, i.e. provide no species of a . According to the feature of inverter, we get logic '1' at the output, which is actually represented by the quantity of species z . Through simulation, we get the quantity

representing logic '1' around 12. To impose logic '1' at the input, in [20], authors claim that an externally-imposed drive is needed to increase the quantity of species a . Based on the strength of the external drive, a will finally get into an equilibrium with a LOW or HIGH signal range. Therefore the transfer curve of the inverter can be drawn by changing the external drive from weak to strong gradually. Here we are not going to analyze the characteristic curve of the inverter because it can be easily done by sweeping the input signal. We are more interested in the transient time behavior of the inverter. So we provide a forced pulse input signal for the inverter as described in the top graph in Fig. 5.1. We simulate the model using STA, and plot the results in the bottom graph in Fig. 5.1. From the waveforms in Fig. 5.1, it can be clearly seen that it is exactly the transient behavior of an inverter. Namely when the input signal is low, the output signal is high and vice versa.



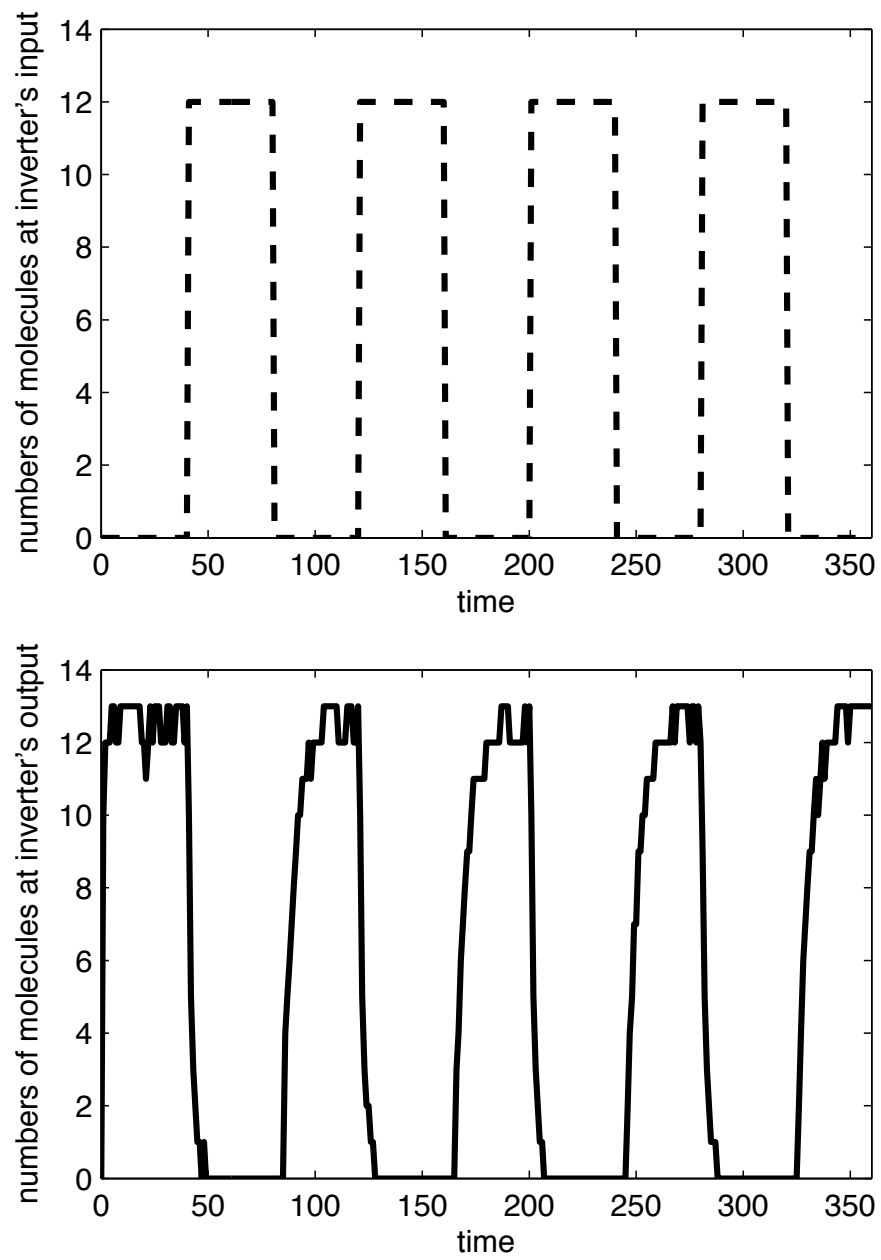
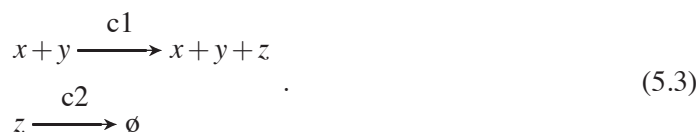


Figure 5.1: Stochastic transient analysis of the INVERTER.

5.2 Some Other Logic Gates

In [20], the approach to create an NAND gate is to "wire-OR" the outputs of multiple inverters by assigning them the same output gene. However no reaction model is given for the NAND gate. So here, we will design some more logic gates with reaction models and use STA to verify our models.

First according to the feature of the AND gate, we design the reaction model of it as in (5.3). The reaction constants $c1$ and $c1$ can be adjusted to get the same logic 1 quantity level as the inverter's. To verify the model, we design the input signals as two forced pulse signals with the same period but different phase, as in the top graph in Fig. 5.2. The simulation result is plotted in the bottom graph in the same figure. As we see from the graphs, the model does exhibit an AND gate. The reaction rates of the AND gate are listed in Table B.5 in Appendix B.



Now we have the AND gate. By simply connecting an inverter to the output of the AND gate, we get an NAND gate. Here the output of the AND gate acts as the external drive for the input of the inverter. To make it clearer, we list the reaction model of NAND gate in (5.2). Please note that in the NAND gate model, the output species of the AND gate is the same as the input species of the inverter. Using the same input signals as for AND gate simulation, we get the simulation results for NAND gate depicted in Fig. 5.3.

We know that an OR relationship can be represented by three NOTs and an AND. Namely if $z = x + y$, we can rewrite it as $z = \overline{\bar{x} \cdot \bar{y}}$. So first we connect x and y to an inverter respectively. Then connect the outputs of the two inverters to an AND gate. Finally connect the output of the AND gate to an inverter again. Then the output of the last inverter is the output of the OR gate, namely z . There are 53 chemical reactions getting involved in the whole OR gate model, which we are not listing here. But we describe the simulation results of the OR gate in Fig. 5.4, using the same input signals as in simulations of AND and NAND gates.

Similar to OR, an XOR relationship, $z = x \oplus y$, can be rewritten as $z = x \cdot \bar{y} + y \cdot \bar{x}$. There are two NOTs, two ANDs and one OR relationships in the XOR model, which consists of total 91 chemical reactions. If we use the same input signals as in the simulations of AND, NAND

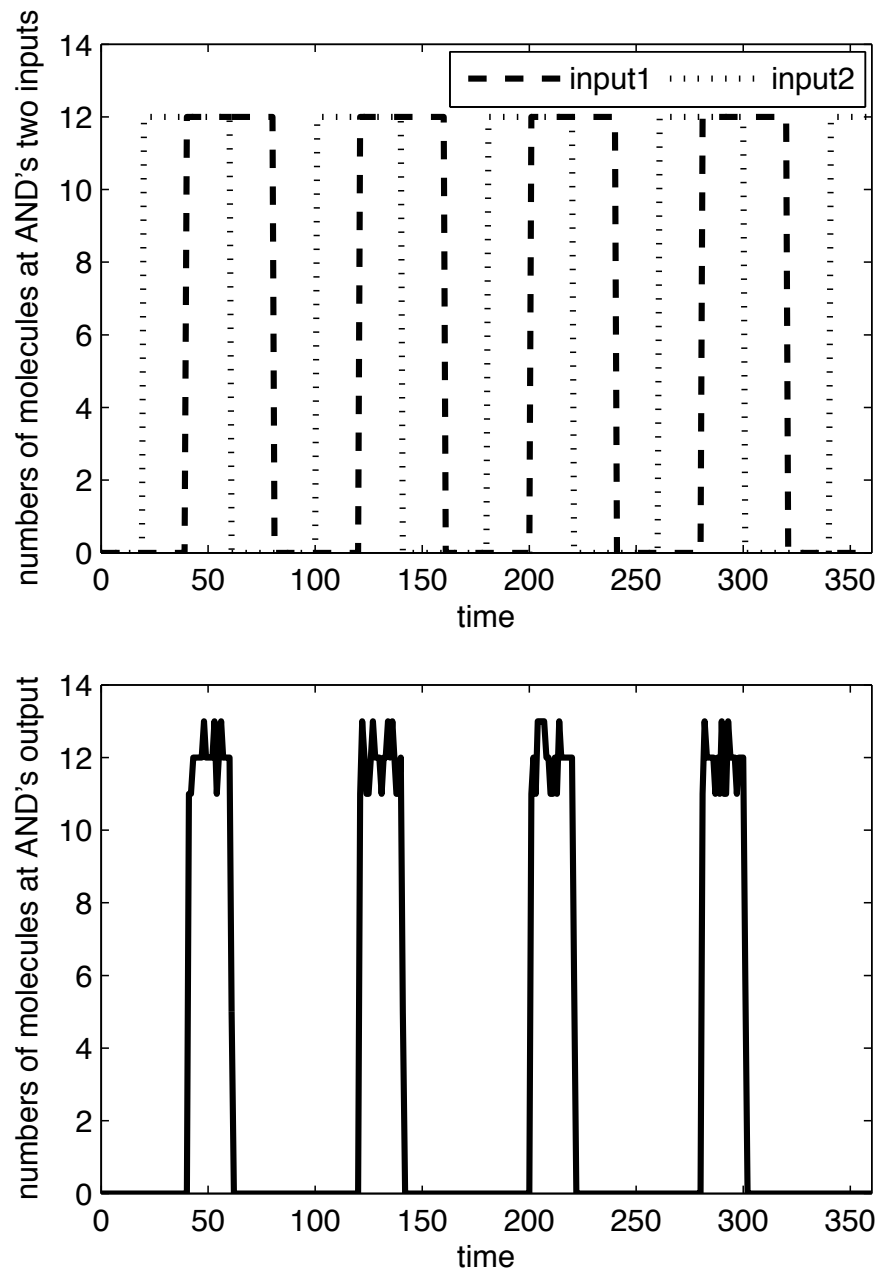


Figure 5.2: Stochastic transient analysis of the AND gate.

and OR gates, we will get the results plotted in Fig. 5.5. From the waveform, we can see that

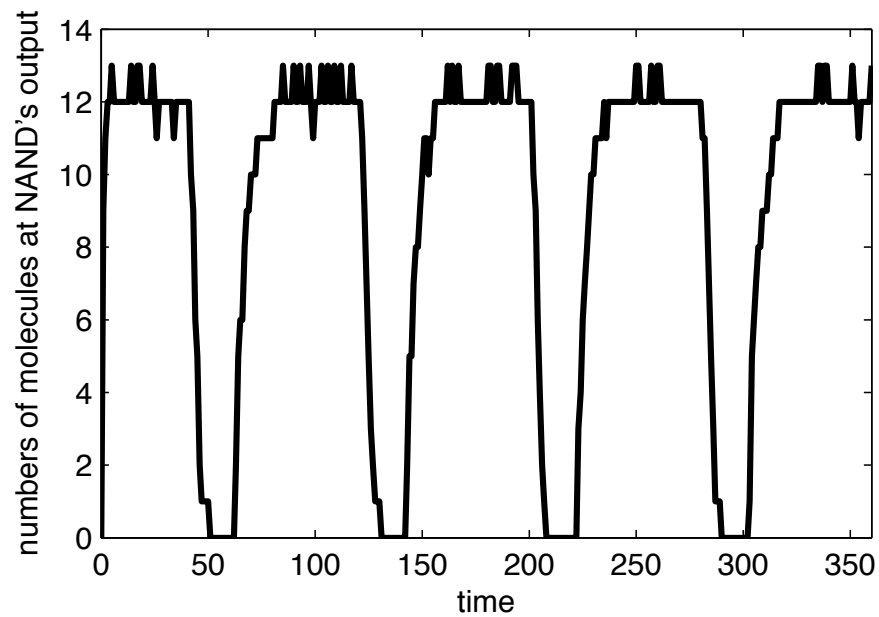


Figure 5.3: Stochastic transient analysis of the NAND gate.

we almost get a mess at the output of XOR gate. The reason why this happens is simply due to the gate delays. It can be easily seen that the output does not have enough time to reach the HIGH level (logic '1') when the input signals change to their next states to make the output go down again. This situation means the frequency of the input signal is too high for the XOR gate to work properly. To verify our judgment here, we slow down the frequency of the input signal to the quarter of the original one and do the simulation again. Now we can see the correct waveform of the XOR gate as plotted in Fig. 5.6.

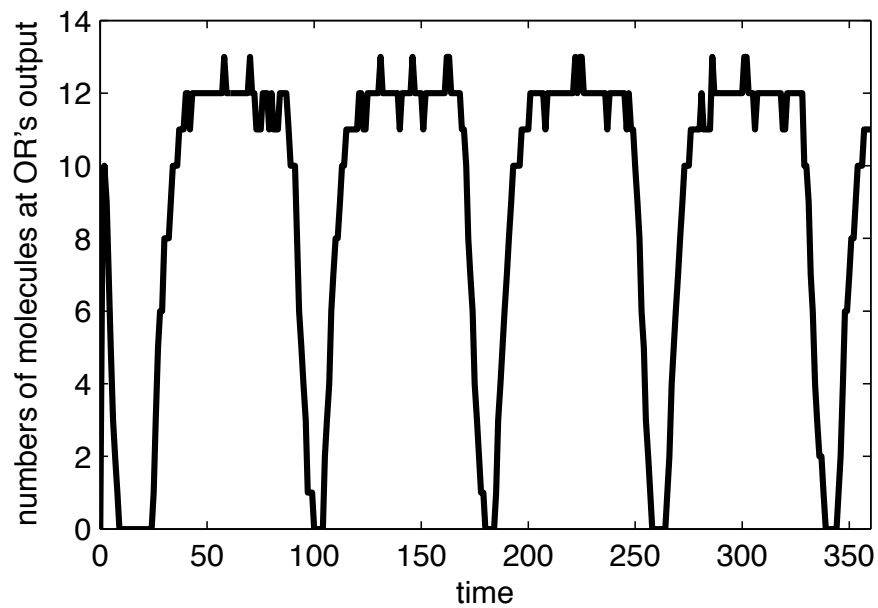


Figure 5.4: Stochastic transient analysis of the OR gate.

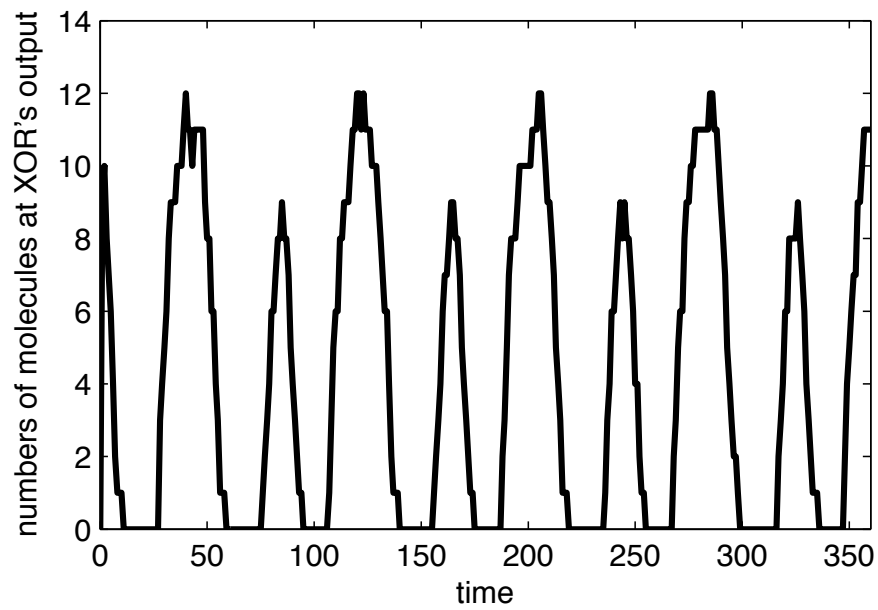


Figure 5.5: Stochastic transient analysis of the XOR gate.

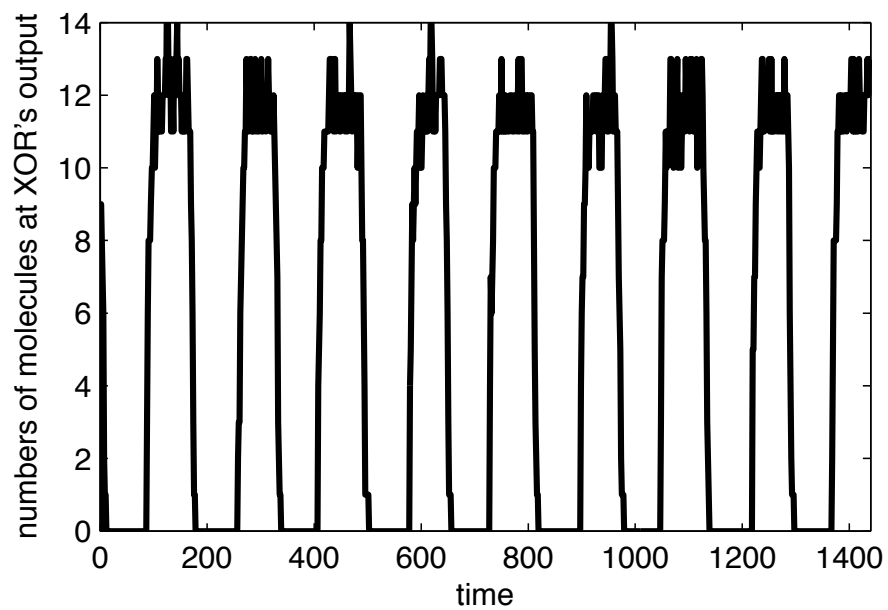


Figure 5.6: Stochastic transient analysis of the XOR gate with lower frequency.

Chapter 6

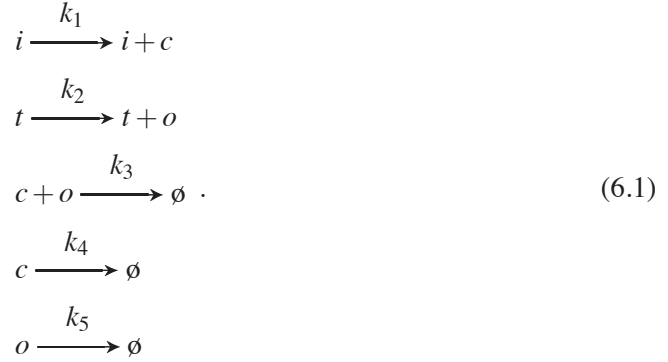
A Simplified Inverter and A 3-stage Ring Oscillator

6.1 Design of A Simplified Inverter

In Chapter 5, we have simulated a biochemical inverter proposed in [20] using STA. The authors of [20] presents the model of the inverter based on the biochemical experiment and analysis. It is quite complex, composed by seventeen chemical reactions. It is not easy to apply such logic element into a digital circuit. And there will be too many reactions even in a small circuit, as we have demonstrated in Chapter 5 that an XOR gate based on such inverter consists of 91 reactions. Moreover, there is no clear approach to adjust the reaction rates in order to get various inverts with different thresholds (the ideal switch point of inverter), which are important for inverters to be capable of fitting into different situations where the numbers of molecules representing logic "1" are different.

In view of the complexity of the inverter presented in [20], we propose a simplified inverter with only five reactions. Our goal here is not to demonstrate how to design a good biochemical inverter, but to show how STA can be applied to do analysis and verification during the design

process. The reaction set of the proposed inverter is listed in (6.1).



In reaction set (6.1), i is the input species, s is the species which represents the threshold of the inverter. The mechanism of the inverter is described as follows: when the quantity of input species i is much less than that of threshold species s , the quantity of product c will be much less than that of product o , as long as $k_1 = k_2$ in the first two reactions. Since c and o are annihilating each other in the third reaction, c will be consumed almost immediately after being created in the first reaction while o will keep increasing in quantity. Of course, k_3 should be equal or close to k_1 and k_2 . The fourth and fifth reactions are for balancing the quantities of c and o , ensuring that both of them would be kept at certain level of quantity if they are dominating in the third reaction. In this case, the fifth reaction causes o to reach equilibrium without keeping increasing. The quantity level in the equilibrium should be the high level representing logic "1". Therefore, with a low level input (logic "0"), a high level output (logic "1") is generated. On the other hand, when the input i is on a high level, c will be dominating in the third reaction. As a result, o will reach a low level.

According to SSA, the firing chance of a reaction is proportional to the product of its reactants and the reaction rate. Thus, if we have $k_1 = k_2 = k_3 = 2k_4 = 2k_5$, and set the threshold of inverter at T , then we will get an inverter with low level quantity approximating zero and high level quantity approximating $2T$. Due to the stochasticity of biochemical reactions and the dynamic interaction between reactions, we are not expecting a perfect inverter with this design. But we do not analyze some characteristics of this inverter here, such as noise margin, transfer curve, delay etc. Those are not our emphasis of this thesis. We want to demonstrate the verification of the inverter using STA.

If we set threshold $T = 50$, and set the reaction rates according to Table B.6 in Appendix

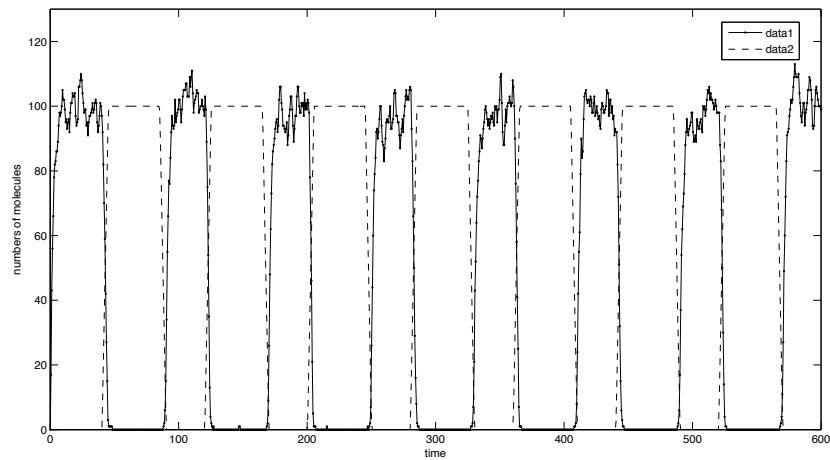


Figure 6.1: A simplified inverter.

B, we can get the stochastic transient simulation result of the inverter shown in Fig. 6.1.

To verify if the logic "1" level varies with threshold, we have T set to 100, and do the simulation again. The result is displayed in Fig. 6.2. It can be clearly seen that the level of logic "1" has reached about 200.

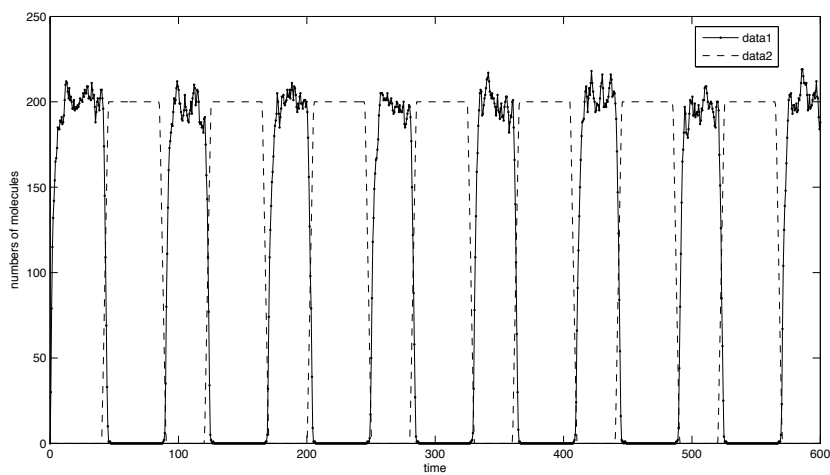


Figure 6.2: A simplified inverter with different threshold.

6.2 Simulation of A 3-stage Ring Oscillator

As we know, if odd number (> 1) of inverters are connected one by one in series and the output of the last inverter is hooked with the input of the first inverter to make a loop, then a ring oscillator can be created, as shown in Fig. 6.3, which has three inverters (is called 3-stage ring oscillator).

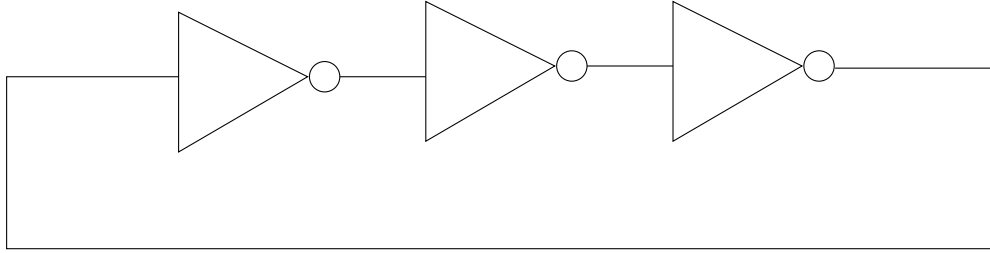
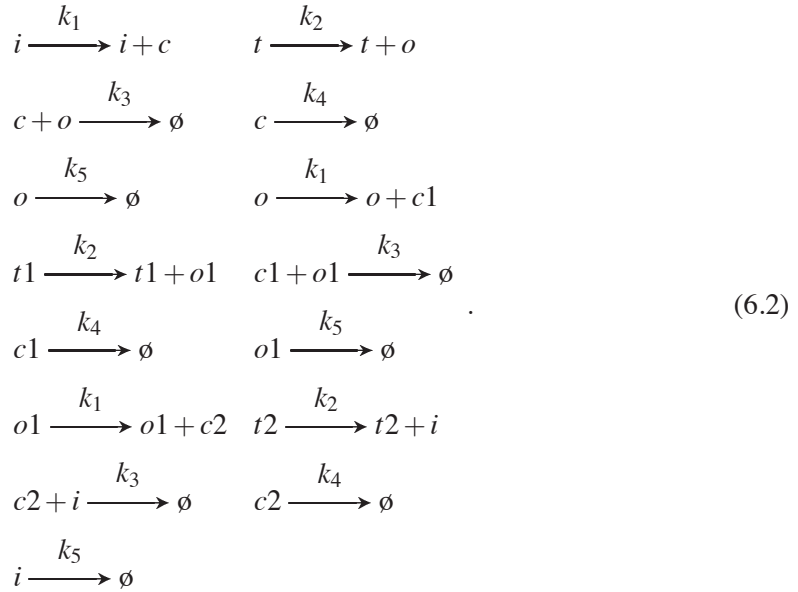


Figure 6.3: A 3-stage ring oscillator.



To verify our design of the simplified inverter, we apply the proposed inverter to build a 3-stage ring oscillator according to Fig. 6.3. The reaction set of the oscillator is shown in (6.2), which is simply composed of three inverters, only the output of the last inverter is the same species as the input of the first one. Simulate the oscillator, we get the waveform displayed in

Fig. 6.4.

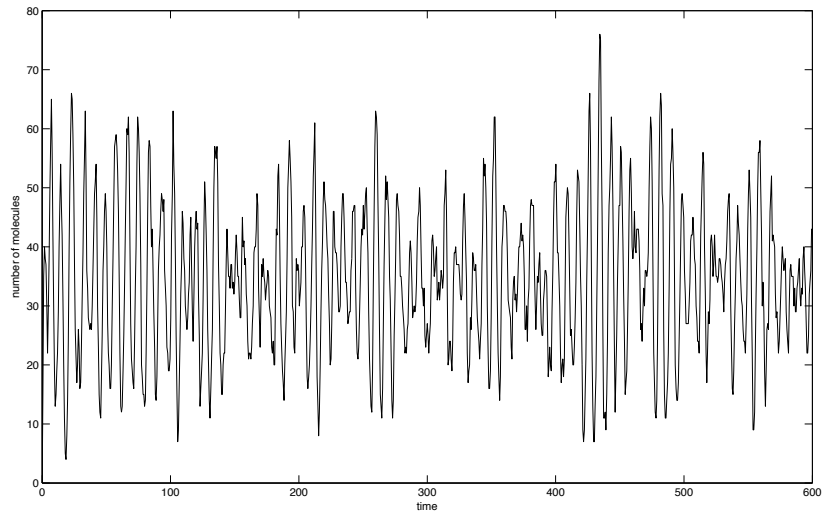
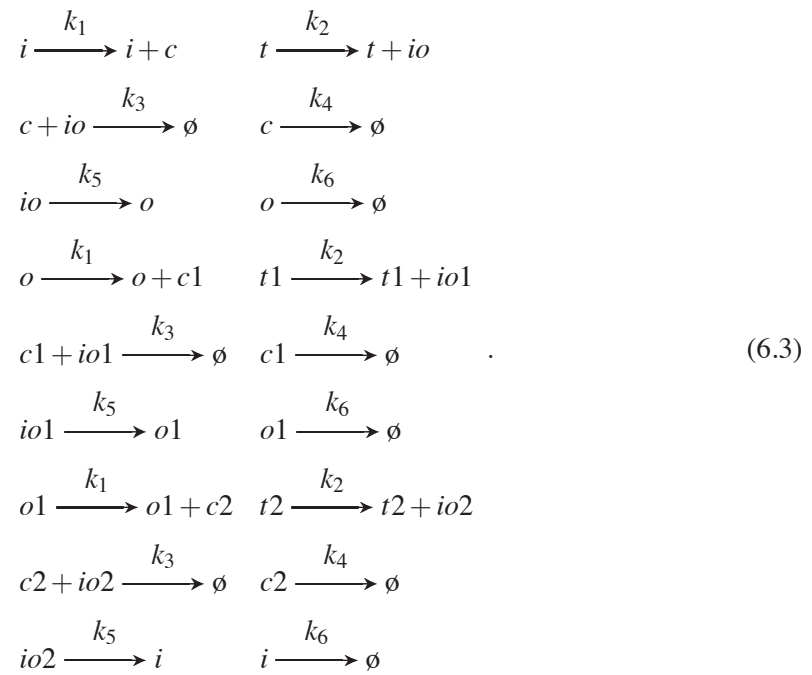


Figure 6.4: A 3-stage biochemical ring oscillator.



Apparently, this is not a good waveform. Due to the small delay of the inverter and stochasticity of the system, the period of it seems too small in terms of the simulation time scale and the amplitude of the waveform varies too much. To make it perform better, a delay is introduced into the inverter and the corresponding reaction set of the oscillator is changed to (6.3). The reaction rates are listed in Table B.7 in Appendix B. Simulate it again, we get the waveform with larger period and much less variation in amplitude, as shown in Fig. 6.5.

Again we want to emphasize that the purpose of presenting our preliminary design here is not to show our capability of designing a simplified biochemical inverter, but to demonstrate that STA can be a powerful analysis and verification tool for biochemical research, especially for biochemical digital circuits.

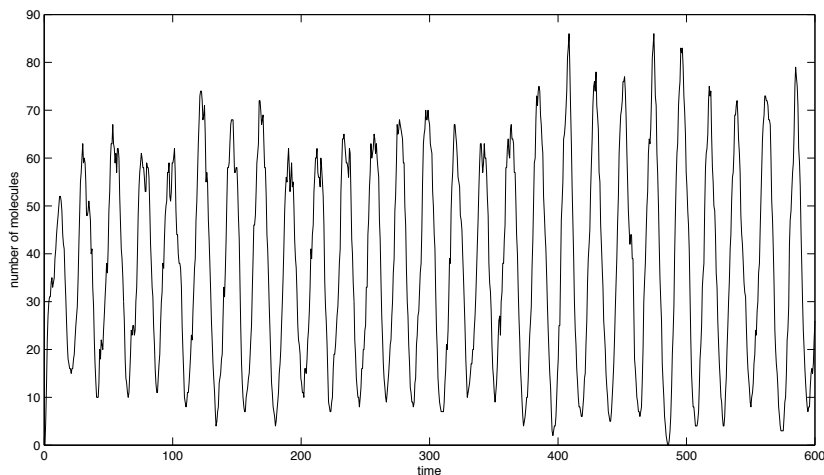


Figure 6.5: A 3-stage biochemical ring oscillator with longer period and less amplitude variation.

Chapter 7

Conclusion and Discussion

Since Gillespie proposed stochastic simulation algorithm for biochemical system about thirty year ago, there hasn't been any systematic way reported to do transient analysis for biochemical system based on Gillespie's SSA. In this thesis, we present such a systematic approach, stochastic transient analysis for biochemical system. We borrow the idea of transient analysis from electrical circuits, classifying the input signals into categories. Then we add extra processing in SSA to get the time response of the system to the input signals. Simulations to Lotka model and some biochemical logic gate designs show that the proposed stochastic transient analysis can yield new insights into the dynamics of biochemical systems. Specifically it can be used to analyze and verify the design of biochemical digital circuits.

As this thesis is focusing on the simulation approach, no further exploration has been done to analyze the delay, noise margin and other features of the designed logic gates or other digital circuits. This can be a research direction in the future, for which STA will act as an indispensable tool. Besides, we are currently working on the incorporation of STA to BioSPICE [19], an open source software package for modeling and simulation of biochemical systems.

Appendix A

Acronyms

A.1 Acronyms

Table A.1: Acronyms

Acronym	Meaning
<i>ODE</i>	Ordinary Differential Equation
<i>DAE</i>	Differential Algebraic Equation
<i>ACM</i>	Association for Computing Machinery
<i>IEEE</i>	Institute of Electrical and Electronics Engineers
<i>SSA</i>	Stochastic Simulation Algorithm
<i>STA</i>	Stochastic Transient Analysis
<i>MNA</i>	Modified Nodal Analysis
<i>PWL</i>	Piece-Wise Linear

Appendix B

Reaction Parameter Tables

B.1 Reaction Parameter Tables

Table B.1: Species Initial Quantities in Lotka Model

Species	Initial Quantity
x	10
y_1	1000
y_2	1000
z	0

Table B.2: Reaction Rates in Lotka Model

Rate	Value
c_1	1
c_2	0.01
c_3	10

Table B.3: Species Initial Quantities of Weiss' Inverter

Species	Initial Quantity
a	0
$a2$	0
z	0
$z2$	0
gz	85
$gza2$	0
$gza4$	0
$mrnaz$	0
$rnap$	8000
$rrna$	1000

Table B.4: Reaction Rates of Weiss' Inverter

Rate	Value
$K_{dim(a)}$	8.333
$K_{sngl(a)}$	0.1667
$K_{dim(z)}$	8.333
$K_{sngl(z)}$	0.1667
$K_{rprs(a2)}$	66.7
$K_{dis(a2)}$	0.2
$K_{rprs(a4)}$	333.3
$K_{dis(a4)}$	0.25
$K_{dec(a)}$	0.5775
$K_{dec(a2)}$	0.5775
$K_{dec(z)}$	0.5775
$K_{dec(z2)}$	0.5775

Continued on next page

Table B.4 – continued from previous page

Rate	Value
$K_{dec(ga2)}$	0.2887
$K_{dec(ga4)}$	0.2887
$K_{dec(mrna)}$	2.0
$K_{xscribe}$	0.0001
K_{xlate}	0.03

Table B.5: Reaction Rates of AND gate

Rate	Value
c_1	1
c_2	12

Table B.6: Reaction Rates of Simplified Inverter

Rate	Value
k_1	1
k_2	1
k_3	1
k_4	0.5
k_5	0.5

Table B.7: Reaction Rates of 3-stage Ring Oscillator

Rate	Value
k_1	1
k_2	1
k_3	1
k_4	0.5
k_5	0.5
k_6	0.5

References

- [1] I. Surovtsova, S. Sable, J. Pahle, and U. Kummer. Approaches to Complexity Reduction in a Systems Biology Research Environment (SYCAMORE). Simulation Conference, 2006. WSC 06. Proceedings of the Winter, pages pp. 1683–1689, 12 2006.
- [2] K. Takahashi, K. Yugi, K. Hashimoto, Y. Yamada, C. Pickett, and M. Tomita. Computational Challenges in Cell Simulation: A Software Engineering Approach. Intelligent Systems, IEEE, 17:no.5, pp. 64–71, 9 2002.
- [3] D. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. Journal of Physical Chemistry, 81:No. 25, pp. 2340–2361, 1977.
- [4] M. Gibson and J. Bruck. Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. Journal of Physical Chemistry, pages 105, 1876–1889, 2000.
- [5] D. Gillespie. Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems. J. Chem. Phys., pages 115(4), 1716–1733, 2001.
- [6] M. Rathinam, Y. Cao, L. Petzold, and D. Gillespie. Stiffness in Stochastic Chemically Reacting Systems: The Implicit Tau-leaping Method. J. Chem. Phys., 2003.
- [7] Y. Cao, D. Gillespie, and L. Petzold. Avoiding Negative Populations in Explicit Tau Leaping. J. Chem. Phys., pages 123, 054104–054112, 2005.
- [8] Y. Cao, D. Gillespie, and L. Petzold. Efficient Stepsize Selection for the Tau-leaping Method. J. Chem. Phys., pages 124, 044109, 2006.

- [9] Y. Cao, D. Gillespie, and L. Petzold. The Adaptive Explicit-implicit Tau-leaping Method with Automatic Tau Selection. J. Chem. Phys., pages 126(22), 224101–224109, 2007.
- [10] B. Fiedler and M. Riedel J. Bruck. Synthesizing Stochasticity in Biochemical Systems. Proceedings of the 44th Annual Conference on Design Automation, pages 640–645, 2007.
- [11] A. Arkin, J. Ross, and H. McAdams. Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in Phage λ -Infected Escherichia Coli Cells. Genetics, pages 149:1633–1648, 1998.
- [12] J. Way D. Drubin and P. Silver. Designing Biological Systems. Genes and Dev, 21:242–254, 2007.
- [13] M. Elowitz and S. Leibler. A Synthetic Oscillatory Network of Transcriptional Regulators. Nature, 403:335–338, 2000.
- [14] L. Nagel. SPICE2: A Computer Program to Simulate Semiconductor Circuits. Memorandum No. UCB/ERL M520. Electronics Research Lab., College of Engi., UC, Berkeley, 5 1985.
- [15] J. May and D. Mikulecky. Glucose Utilization in Rat Adipocytes. Journal of Biological Chemistry, pages 258, No. 8, pp. 4881–4888, 4 1983.
- [16] R. Seither, D. Hearne, D. Trent, D. Mikulecky, and D. Goldman. SPICE2 Network Simulation of Antifolate Effects on Purine and Pyrimidine Biosynthesis: Exploring the Role of Tetrahydrofolate Cofactor Depletion Versus Dihydrofolate Feedback Inhibition. Computera and Mathematics with Applications, pages 20, No. 4–6, pp. 88–101, 1990.
- [17] R. Seither, D. Trent, D. Mikulecky, T. Rape, and D. Goldman. Effect of Direct Suppression of Thymidylate Synthase at the 5,10-Methylenetetrahydrofolate Binding Site on the Interconversion of Tetrahydrofolate Cofactors to Dihydrofolate by Antifolates. Journal Of Biological Chemistry, pages 266, No. 8, pp. 18016–18023, 3 1991.
- [18] A. Brower, R. Ewing, R. Brower, and H. Abdel-Aty-Zohdy. SPICE for Modeling Biochemical Networks. Circuits and Systems, 2005. 48th Midwest Symposium on, 1:491–494, 8 2005.

- [19] S. Kumar and J. Feidler. BioSPICE: A Computational Infrastructure for Integrative Biology. OMICS: A Journal of Integrative Biology, pages 7(3): 225–225, 9 2003.
- [20] R. Weiss, G. Homsy, and T. Knight Jr. Toward In-vivo Digital Circuits. In Dimacs Workshop on Evolution as Computation Princeton NJ, 1 1999.
- [21] 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, 2:47–84, 2003.
- [22] M. Lauria, K. Bhalerao, M. Pugalanthiran, and B. Yuan. Building Blocks of A Biochemical CPU Based on DNA Transcription Logic. 3rd Workshop on Non-Silicon Computation(NSC-3),Munich, 6 2004.