A Mathematical Model of a Denitrification Metabolic Network in *Pseudomonas aeruginosa*

Seda Arat

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Reinhard C. Laubenbacher, Chair
John A. Burns
M. Stanca Ciupe

November 30th, 2012
Blacksburg, Virginia

Keywords: Systems Biology, Dynamical Systems, Discrete Models

A Mathematical Model of a Denitrification Metabolic Network in *Pseudomonas aeruginosa*

Seda Arat

ABSTRACT

Lake Erie, one of the Great Lakes in North America, has witnessed recurrent summertime low oxygen dead zones for decades. This is a yearly phenomenon that causes microbial production of the greenhouse gas nitrous oxide from denitrification. Complete denitrification is a microbial process of reduction of nitrate to nitrogen gas via nitrite, nitric oxide, and greenhouse gas nitrous oxide. After scanning the microbial community in Lake Erie, *Pseudomonas aeruginosa* is decided to be examined, not because it is abundant in Lake Erie, but because it can perform denitrification under anaerobic conditions. This study focuses on a mathematical model of the metabolic network in *Pseudomonas aeruginosa* under denitrification and testable hypotheses generation using polynomial dynamical systems and stochastic discrete dynamical systems. Analysis of the long-term behavior of the system changing the concentration level of oxygen, nitrate, and phosphate suggests that phosphate highly affects the denitrification performance of the network.

This work was partially supported in part by the Ministry of Education in Turkey.

Acknowledgments

I am very grateful to my advisor, Reinhard Laubenbacher, for the help, encouragement, and positive reinforcement I have received throughout my studies. I thank the committee members John Burns and Stanca Ciupe, and my collaborators, George Bullerjahn, Michael Schlais, David Murrugarra, Alan Veliz-Cuba, and Boris Aguilar for their support, and permission to include material.

I thank everyone in the Department of Mathematics and at the Virginia Bioinformatics Institute at Virginia Tech for providing an excellent international and interdisciplinary environment. I'm especially thankful to Peter Haskell, Nick Loehr, and Nicole Sutphin along with former and current members of my research group: David, Franziska, Shernita, Madison, Claus, Matt, Kasia, Anael, Cory, Jennifer, Kathy, and Betsy. I would like to thank the professors at the University of Pennsylvania and Ankara University who gave me a strong education foundation in order to pursue a doctoral degree in mathematics, especially Max Mintz, Ali Bulent Ekin, and Ethem Derman for their invaluable advice and friendship.

I thank my friends, who make my life more enjoyable, are scattered all over the world: Onur, Burcu and Altug, Vitor, Hans, Weiwei, Alex, Bharat, Shiva, Vikas, Kunjin, Nil, Cigdem, and Malik. I give warm thanks to Shernita and Alyse for the time dedicated to the proofreading of this thesis.

I would like to thank my husband, Ali; his love, embrace, and support kept me standing up whenever I faced difficulty. Last but not least, I am deeply grateful to my parents, for without their love and support I would not be here, and my *dear* grandma, who took care of me when I was a kid, and my brother who makes me laugh out loud.

Dedication

Dedicated to *my family*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Mathematics is a gate and key to the sciences."

Roger Bacon (1214-1294)

Systems biology is an interdisciplinary field of study that aims to discover functionality of the complex interactions between the components of a biological system of the interest, and model the dynamic behavior of the system. The system of the interest can be a gene regulatory network, a signaling network, or a *metabolic network*, which is a set of genes, enzymes, proteins, and metabolites, and the chemical and regulatory reactions between them. One of the fundamental instruments of systems biology is *mathematical modeling*, which consists of a system of equations representing the interactions among the variables in the system. There are several reasons for the necessity of mathematical models such as:

- systematizing data related to the system of interest [1],
- comprehending the essential features of the underlying system [2],
- paving the way of computer simulation of the system of interest by utilizing the intensive numerical and mathematical analysis [3],

- suggesting testable hypotheses, and predicting their outcomes [4],
- identifying and correcting misinterpreted results [1].

A mathematical model is an analog of a real world system. In order to declare whether the model is valid, the accuracy of its predictions must be tested [4]. If it is invalid, then the model should be fine-tuned and make new predictions until the model becomes valid. The model validation process is called *systems biology approach*, which is extensively used especially in biochemistry and biomedicine [1].

There are various ways of dynamic modeling of a system in the literature such as a system of *differential equations*, which is a continuous state and time model, a *discrete dynamical system* and *agent-based model*, which are state and time discrete models, and a *continuous time Boolean model* and *state discrete Markov chain*, which are hybrid models, and their stochastic variants.

A modeler first should determine what the problem is and why they need a model [1], and then decide which details the model should include and which feature(s) of the biological system should be captured before deciding the model type [5]. My project aims to discover environmental factors yielding increased production of the greenhouse gas nitrous oxide (N_2O) in Lake Erie during summertime oxygen depletion through the examination of the metabolic network of *Pseudomonas aeruginosa* during denitrification. A mathematical model is needed to organize information about hypoxia in Lake Erie, the denitrification process, and the microbe, *P. aeruginosa*. The project goal is to incorporate these topics and design a dynamic computer model to simulate, and generate hypotheses that my collaborator, Dr. George Bullerjahn, can test in his lab at Bowling Green State University in Bowling Green, OH. An extensive literature search led to the determination of the model variables, which are regulatory proteins with an important role in denitrification, denitrification enzymes, nitrogen oxides, and their interactions. This study intends to capture the effect of phosphate (PO_4) on the main regulator of the system, as well as the whole system using my model which incorporates the systems biology approach. This study can be considered as under

the *systems microbiology* umbrella, an interdisciplinary field of study intersecting systems biology and microbiology. Lastly, I chose the state and time discrete dynamical systems as a modeling strategy for this study due to the following reasons:

- discrete models tend to be intuitive and accessible to life scientists [6],
- continuous models use kinetic parameters, and estimation of these parameters requires detailed information about the system, which is not always possible; whereas, discrete models do not use kinetic parameters,
- discrete systems can be used to approximate numerical solutions of continuous systems, most of which can not be solved analytically.

The thesis is structured as follows: Chapter 2 consists of *discrete dynamical systems*, and its deterministic and stochastic variations such as finite dynamical systems, polynomial dynamical systems (PDS), and stochastic discrete dynamical systems (SDDS). Chapter 3 provides biological background about dead zones, Lake Erie, *Pseudomonas aeruginosa*, and *initial* (deterministic and stochastic) models of denitrification metabolic network in *Pseudomonas aeruginosa*. Finally, Chapter 4 includes the summary of the study, and the future work.

The paper that I coauthored and discussed in this thesis is "Modeling Stochasticity and Variability in Gene Regulatory Networks" by David Murrugarra, Alan Veliz-Cuba, Boris Aguilar, **Seda Arat**, Reinhard Laubenbacher. It was published in EURASIP Journal on Bioinformatics and Systems Biology in 2012. My contributions to the paper were the design of the algorithms, and coding the stochastic framework described in the paper, which is available at http://dvd.vbi.vt.edu/adam.html.

Chapter 2

Discrete Dynamical Systems (DDS)

Multi-state and time discrete models are widely accepted in systems biology, not because they are highly intuitive, but because they provide coarse-grained view of the structure and key dynamical features of biological networks such as bistability, steady states, and limit cycles. In the general setting, a network is a directed graph whose nodes represent genes, proteins, or other molecular components, and whose edges represent biological interactions between these nodes (i.e. activation/inhibition). If there are n nodes in the network, a *state* is an n-dimensional vector that consists of values of the nodes. The biological interactions between these nodes can be written as logical rules.The next state of a current state can be computed by these logical rules. The *state space* is a directed graph whose nodes are all possible states and edges are transitions between the states.

In this chapter, three state and time discrete modeling frameworks are discussed: finite dynamical systems, polynomial dynamical systems, and stochastic discrete dynamical systems. Both finite dynamical systems and polynomial dynamical systems are deterministic, i.e. there is a single next state with probability 1 for a given current state; whereas, stochastic discrete dynamical systems allow more than one next state with some probability less than 1 for a given current state. Besides, the *Markov property* holds, the next state depends upon only the current state. The knowledge of previous states does not provide any additional information about the next state [7].

Definition 2.0.1. Let $x_1, x_2, \ldots, x_n$ be nodes (variables) which can take values in finite sets $X_1, X_2, \ldots, X_n$ respectively, and $X = X_1 \times \cdots \times X_n$ be the Cartesian product. A *discrete dynamical system* is a function

$$f = (f_1, f_2, \ldots, f_n) : X \to X$$

where an update (coordinate) function of x_i, $f_i : X \to X_i$ is a function in a subset of $\{x_1, x_2, \ldots, x_n\}$ for $i = 1, 2, ..., n$.

Definition 2.0.2. $x = (x_1, x_2, \ldots, x_n)$ is called a *state* of a system.

Definition 2.0.3. Let $x(t)$ be the current state of a system. Then, $x(t+1)$ is called the *next state* of the current state, $x(t)$, and $x(t+1) = f(x(t))$.

Definition 2.0.4. Let $x = (x_1, x_2, \ldots, x_n)$, and $f = (f_1, f_2, \ldots, f_n)$ where n is the number of variables in the system. Then, x is called a *steady state* (or *fixed point*) of the system if $f(x) = x$.

2.1 Finite Dynamical Systems (FDS)

A finite dynamical system is a discrete dynamical system with a finite number of nodes (variables). It is commonly used to model regulatory networks in systems biology [8].

Example 2.1.1. Let $n = 2, X = \{0, 1\} \times \{0, 1\}$. All possible states are (0, 0), (0, 1), (1, 0), and (1, 1). Table 2.1 and 2.2, which are called *transition tables*, represent which variable(s) affect the update rule of x_1 and x_2, respectively.

Table 2.1: Transition table for the variable x_1 in Example 2.1.1, 2.2.1, and 2.3.1

$x_1(t)$	$x_2(t)$	$x_1(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

Table 2.2: Transition table for the variable x_2 in Example 2.1.1, 2.2.1, and 2.3.1

$x_1(t)$	$x_2(t+1)$
0	1
1	0

Note that in Table 2.1, $f_1(x(t)) = x_1(t+1)$ is 0 when the values of $x_1(t)$ and $x_2(t)$ are the same, and is 1 when the values of $x_1(t)$ and $x_2(t)$ are different. Then, an update function of x_1 is

$$f_1(x) = \begin{cases} 0 & \text{if } x_1 = x_2 \\ 1 & \text{if } x_1 \neq x_2 \end{cases}$$

Similarly, note that in Table 2.2, $f_2(x(t)) = x_2(t+1)$ is 0 when the value of $x_1(t)$ is 1, and is 1 when the value of $x_1(t)$ is 0 regardless of the value of $x_2(t)$. Then, an update function of x_2 is

$$f_2(x) = 1 - x_1$$

Then, the Finite Dynamical System in the variables x_1, x_2 is

$$f = (f_1, f_2) : X \to X.$$

Figure 2.1: Wiring Diagram (Dependency Graph) of the network in Example 2.1.1, 2.2.1, and 2.3.1

Figure 2.1 is the network corresponding to the system. It can also be called a *wiring diagram* or a *dependency graph*. Both x_1 and x_2 are needed for the update function of x_1 whereas only x_1 is needed for the update function of x_2.

Many discrete networks can be modeled as polynomial dynamical systems, which are multi-state and time discrete dynamical systems that use polynomial functions over a finite field. The main advantage of polynomial dynamical systems is a unifying algebraic framework for many discrete modeling types [6]. Since PDS allows the use of mathematical tools from computational algebra, it is easily programmable for model construction and analysis of discrete systems with polynomial time complexity, i.e. running time is not slower than n^k for some constant k, where n is the number of nodes (variables) in the system [9].

2.2 Polynomial Dynamical Systems (PDS)

Theorem 2.2.1 (Lidl et al. [10])**.** Let $\mathbb{F}$ be a finite field, and $h : \mathbb{F}^n \to \mathbb{F}$. Then there exists a polynomial $f : \mathbb{F}^n \to \mathbb{F}$ such that $h(x) = f(x)$ for all $x \in \mathbb{F}^n$.

Definition 2.2.1 (Laubenbacher et al. [9])**.** Let $\mathbb{F} = F_p$ be a finite field where p is prime, and $x_1, x_2, \ldots, x_n$ be variables in $\mathbb{F}$. A *polynomial dynamical system* is a multi-state and time discrete dynamical system

$$f = (f_1, f_2, \ldots, f_n) : \mathbb{F}^n \to \mathbb{F}^n$$

where an update function of x_i, f_i is a polynomial in a subset of $\{x_1, x_2, \ldots, x_n\}$ for $i = 1, 2, \ldots, n$.

In light of Theorem 2.2.1, we can find an update function of a variable x_i by using the polynomial form:

$$f_i(x) = \sum_{(c_{i_1},\ldots,c_{i_r})\in\mathbb{F}^r} f_i(c_{i_1}, \ldots, c_{i_r}) \prod_{j\in\{i_1,\ldots,i_r\}} (1 - (x_j - c_j)^{p-1}) \pmod{p} \tag{2.1}$$

where $c_{i_1}, \ldots, c_{i_r}$ are the values of the variables $x_{i_1}, \ldots, x_{i_r}$, which affect the update of x_i in the transition table for x_i, $\mathbb{F} = F_p = \{0, 1, \ldots, p-1\}$, and p is the maximum (prime) number of different discrete values that all variables can take on [6].

In Definition 2.2.1, it is assumed that the number of states each variable can take on is a prime number. It can be achievable by discretizing a time-continuous data into a finite set. There are several ways for data discretization; one method is to use thresholds. For instance, the network becomes Boolean if there is only one threshold for each gene. If the expression level of a gene is below its threshold, then the gene is considered inactive, and its (discretized) value is set to 0. Otherwise, the gene is considered active, and its (discretized) value is set to 1. The choice of p can depend upon the availability of data, the level of knowledge of the biology behind the gene interactions, the number of genes in the network, and/or the connectivity of the network [9].

Example 2.2.1. Let $n = 2, p = 2, \mathbb{F} = F_2$, and Table 2.1 and 2.2, which are called the transition tables, represent which variable(s) affect the update rule of x_1 and x_2, respectively. Figure 2.1 is the wiring diagram (or dependency graph) of the network. Note that both x_1 and x_2 are inputs for the transition table of x_1; whereas, x_1 is the only input for the transition table of x_2. Then, an

update function of x_1 is

$$f_1(x) = \sum_{(c_1,c_2)\in\mathbb{F}^2} f_1(c_1,c_2) \prod_{j\in\{1,2\}} (1-(x_j-c_j)^{2-1}) \pmod 2$$

$$\begin{aligned}
&= f_1(0,0) * (1-(x_1-0))(1-(x_2-0)) + \\
&\quad f_1(0,1) * (1-(x_1-0))(1-(x_2-1)) + \\
&\quad f_1(1,0) * (1-(x_1-1))(1-(x_2-0)) + \\
&\quad f_1(1,1) * (1-(x_1-1))(1-(x_2-1)) \pmod 2 \\
&= 0 * (1-x_1)(1-x_2) + 1 * (1-x_1)(2-x_2) + 1 * (2-x_1)(1-x_2) + \\
&\quad 0 * (2-x_1)(2-x_2) \pmod 2 \\
&= -x_1 - x_2 + 2x_1x_2 \pmod 2 \\
&= x_1 + x_2
\end{aligned}$$

Similarly, an update function of x_2 is

$$f_2(x) = \sum_{(c_1)\in\mathbb{F}^1} f_2(c_1) \prod_{j\in\{1\}} (1-(x_j-c_j)^{2-1}) \pmod 2$$

$$\begin{aligned}
&= f_2(0) * (1-(x_1-0)) + f_2(1) * (1-(x_1-1)) \pmod 2 \\
&= 1 * (1-x_1) + 0 * (2-x_1) \\
&= 1 - x_1 \pmod 2 \\
&= x_1 + 1
\end{aligned}$$

Thus, the PDS in the variables x_1, x_2 is

$f = (f_1, f_2) : \mathbb{F}^2 \to \mathbb{F}^2$ where $f_1(x) = x_1 + x_2$, and $f_2(x) = x_1 + 1$.

Figure 2.2: State space of Example 2.2.1

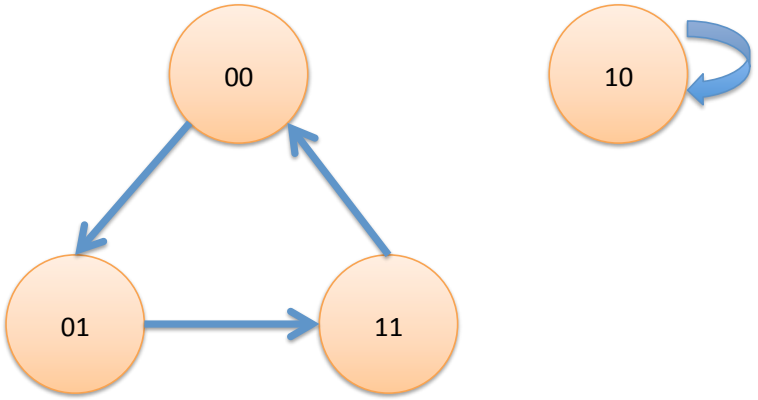


Figure 2.2 depicts the state space of the system, which is a directed graph whose nodes are states (x_1, x_2), and whose edges are transitions. (0, 0) transitions to (0, 1), (1, 1), and back to (0, 0). This is called length-3 limit-cycle. (1, 0) is the only steady state in the system.

A polynomial dynamical system is a mathematical framework for deterministic systems. It is well known that (microbial) regulatory networks are highly stochastic [11]; there is some probability that the variables may not be updated due to stochastic effects. A stochastic discrete dynamical system is a time and state discrete dynamical system modeling stochasticity at the function level.

2.3 Stochastic Discrete Dynamical Systems (SDDS)

Definition 2.3.1 (Murrugarra et al. [12])**.** Let $x_1, x_2, \ldots, x_n$ be variables which can take values in finite sets $X_1, X_2, \ldots, X_n$ respectively. Let $X = X_1 \times \cdots \times X_n$ be the Cartesian product, and $f_i : X \to X_i$ be an update function for x_i for all $i = 1, 2, \ldots, n$. A *stochastic discrete dynamical system* is a collection of n triplets

$$\{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$$

where $p_i^{\uparrow}$ is the activation propensity and $p_i^{\downarrow}$ is the degradation propensity of the i-th variable with $p_i^{\uparrow}, p_i^{\downarrow} \in [0,1]$ for $\forall i \in \{1,2,\ldots,n\}$.

There are two different propensities in this definition. The activation propensity ($p_i^{\uparrow}$) indicates how likely the value of x_i will be increased based on its update function. The degradation propensity ($p_i^{\downarrow}$) indicates how likely the value of x_i will be decreased based on its update function. The propensities can be considered as analogies of activation and degradation kinetic rates in differential equations modeling. If all propensity parameters are 0.5, then each variable has a 50% chance to be updated, i.e. the system will be randomly updated. If all propensity parameters are 1, then all variables will be updated with probability 1, i.e. the system becomes deterministic. Interestingly, the steady state(s) of the network is always the same no matter which setting is used; however, the structure of the state space may be slightly changed due to propensity.

In order to obtain transition probabilities, which indicate how probable any given state transitions to another state, let $x(t) = (x_1(t), x_2(t), ..., x_i(t), ..., x_n(t))$ be the current state, $x(t+1) = (x_1(t+1), x_2(t+1), ..., x_i(t+1), ..., x_n(t+1))$ be the next state of the current state in the transition table for all states, and $y = (y_1, y_2, ..., y_i, ..., y_n)$ be a possible future state of the current state. Algorithm 2.3.1 demonstrates how a transition probability of each coordinate of a state can be computed. If y_i is equal to $x_i(t+1)$, and greater (or less) than $x_i(t)$, then the probability that $x_i(t)$ will be updated is the activation (or degradation) propensity. If y_i is equal to $x_i(t)$ and $x_i(t+1)$, then $x_i(t)$ will stay at its current state with probability 1. However, the probability that $x_i(t)$ will be updated is 0 if y_i is not equal to either $x_i(t)$ or $x_i(t+1)$.

Algorithm 2.3.1: TRANSITION PROBABILITY($x_i(t), x_i(t+1), y_i$)

comment: Find the probability of transition of $x_i(t)$ to y_i

$p \leftarrow 0$

if $y_i = x_i(t+1)$

then
- **if** $x_i(t) < x_i(t+1)$
  - **then** $p \leftarrow p_i^{\uparrow}$
- **else if** $x_i(t) > x_i(t+1)$
  - **then** $p \leftarrow p_i^{\downarrow}$
- **else** $p \leftarrow 1$

else if $y_i = x_i(t)$

then
- **if** $x_i(t) < x_i(t+1)$
  - **then** $p \leftarrow 1 - p_i^{\uparrow}$
- **else if** $x_i(t) > x_i(t+1)$
  - **then** $p \leftarrow 1 - p_i^{\downarrow}$
- **else** $p \leftarrow 1$

return (p)

Algorithm 2.3.1 computes only the probability of transition of $x_i(t)$ to y_i. In order to find the transition probability from a current state ($x(t)$) to a possible future state (y), this algorithm must run n times where n is the number of variables in the system, and all obtained probabilities must be multiplied. The result is the probability with which the current state transitions to a possible future state. Transition probabilities are nonnegative real numbers between 0 and 1.

A transition probability matrix is an $n \times n$ matrix that describes the probability of state transitions for a given system. The (j, k)-th entry of the transition matrix is the transition probability that the j-th state transitions to the k-th state. The starting state is the j-th state and the possible future state is the k-th state. The transition matrix of a system is not necessarily symmetric.

Stochastic discrete dynamical systems (SDDS) can be considered as polynomial dynamical systems with propensity parameters. Therefore, an SDDS modeler can use polynomials for update functions over a finite field F_p where p is the maximum prime number that all variables can take on.

Example 2.3.1. Let $n = 2, X = \{0, 1\} \times \{0, 1\}$, Table 2.1 and 2.2, which are called the transition tables, represent which variable(s) affect the update rule of x_1 and x_2, respectively, and Table 2.3 represent the activation and degradation propensity parameters of x_1 and x_2. The wiring diagram (or dependency graph) of the system is shown in Figure 2.1. The transition table for all states (Table 2.4) can be constructed using Table 2.1 and 2.2.

By Example 2.2.1, the update function of x_1 and x_2 are $f_1(x) = x_1 + x_2$, and $f_2(x) = x_1 + 1$, respectively. Then, the SDDS in the variables x_1, x_2 is

$$(\{f_1, 0.3, 0.5\}, \{f_2, 0.6, 0.8\}).$$

Table 2.3: Propensity parameters for the variables in Example 2.3.1

	x_1	x_2
Activation	0.3	0.6
Degradation	0.5	0.8

Table 2.4: Transition table for all states in Example 2.3.1

$x_1(t)$	$x_2(t)$	$x_1(t+1)$	$x_2(t+1)$
0	0	0	1
0	1	1	1
1	0	1	0
1	1	0	0

Algorithm 2.3.1 is used to find the transition probabilities. Since there are two variables in the system, the algorithm must run twice to obtain the transition probability of a current state to a possible future state. For example, suppose the current state is $x(t) = (0, 0)$, and a possible future state is $y = (0, 0)$. By Table 2.4, the next state of the current state in Table 2.4 is $x(t+1) = (0, 1)$. Note that the first coordinate of the current state ($x_1(t)$) is 0, the first coordinate of the possible future state (y_1) is 0, and the first coordinate of the next state ($x_1(t+1)$) is 0. Since all are the same, the probability is 1. Similarly, in order to find the transition probability of the second coordinate, the second coordinate of the current state ($x_2(t)$) is 0, the second coordinate of the possible future state (y_2) is 0, and the second coordinate of the next state ($x_2(t+1)$) is 1. Since y_2 is equal to $x_2(t)$, and less than $x_2(t+1)$, the probability is $1 - p_2^{\uparrow}$. Hence, the transition probability of (0, 0) to (0, 0) is $1*(1-0.6) = 0.4$. Other transition probabilities can be calculated in the same manner.

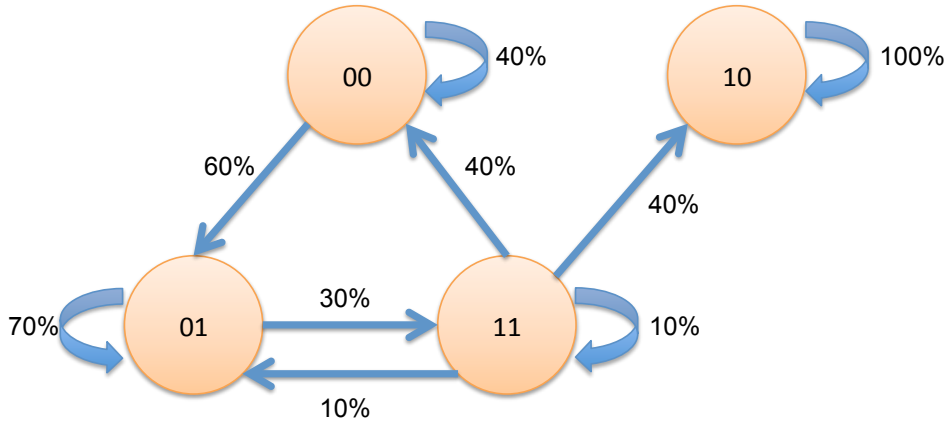
- $Pr(00 \to 00) = (1.0) * (1 - 0.6) = 0.4$, $Pr(00 \to 01) = (1.0) * (0.6) = 0.6$
  $Pr(00 \to 10) = (0.0) * (1 - 0.6) = 0.0$, $Pr(00 \to 11) = (0.0) * (0.6) = 0.0$
- $Pr(01 \to 00) = (1 - 0.3) * (0.0) = 0.0$, $Pr(01 \to 01) = (1 - 0.3) * (1.0) = 0.7$
  $Pr(01 \to 10) = (0.3) * (0.0) = 0.0$, $Pr(01 \to 11) = (0.3) * (1.0) = 0.3$
- $Pr(10 \to 00) = (0) * (1) = 0$, $Pr(10 \to 01) = (0) * (0) = 0$
  $Pr(10 \to 10) = (1) * (1) = 1$, $Pr(10 \to 11) = (1) * (0) = 0$
- $Pr(11 \to 00) = (0.5) * (0.8) = 0.4$, $Pr(11 \to 01) = (0.5) * (1 - 0.8) = 0.1$
  $Pr(11 \to 10) = (1 - 0.5) * (0.8) = 0.4$, $Pr(11 \to 11) = (1 - 0.5) * (1 - 0.8) = 0.1$

Table 2.5 is the transition probability matrix, which can be considered as a limiting distribution of the system. Figure 2.3 is the state space of the system, which illustrates all possible trajectories to follow from any given initial state of the network. The numbers adjacent to the edges are the transition probabilities. Edges with probability 0 are not shown. Since this system is not deterministic,

Table 2.5: Transition probability matrix for all possible states in Example 2.3.1

	00	01	10	11
00	0.4	0.6	0.0	0.0
01	0.0	0.7	0.0	0.3
10	0.0	0.0	1.0	0.0
11	0.4	0.1	0.4	0.1

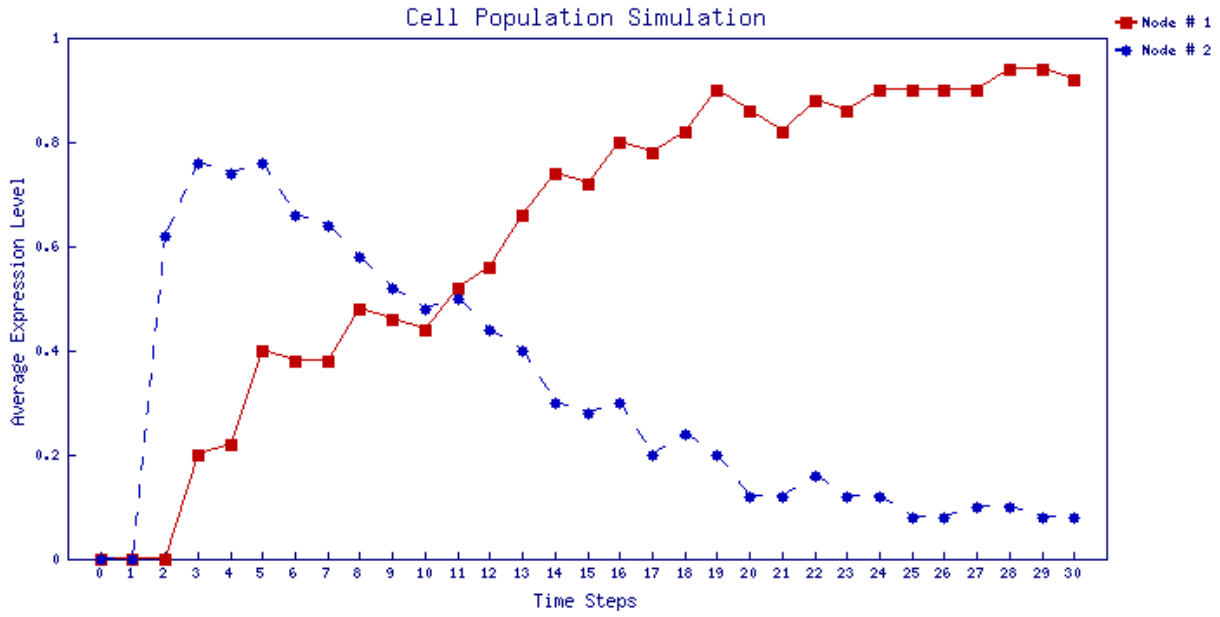
Figure 2.3: State space of Example 2.3.1



most states have different trajectories. For instance, (0, 0) transitions to (0, 1) in the deterministic system; whereas, (0, 0) transitions to (0, 1) with probability 0.6, and stays at its own state with probability 0.4 in the stochastic system. Similarly, (1, 1) transitions to (1, 0) in the deterministic system; whereas, (1, 1) transitions to (0, 0) and (1, 0) with probability 0.4, and transitions to (0, 1) with probability 0.1, and stays at its own state with probability 0.1. The system still has only one steady state, which is (1, 0); however, the system no longer has the length-3 limit cycle.

With 50 simulations, 30 time steps, and (0, 0) as an initial state, Figure 2.4 shows the long-term behavior of x_1 and x_2. We can also figure out the steady state (fixed point) of the system from Figure 2.4. As time increases, the first node (x_1) approaches 1 when the second node (x_2) approaches

Figure 2.4: Population dynamics of all variables in Example 2.3.1



0. Therefore it can be concluded that (1, 0) is a potential steady state.

Analysis of Dynamic Algebraic Models (ADAM), a free online software tool to analyze the dynamics of discrete networks, available at http://dvd.vbi.vt.edu/adam.html [13], can be used to test all examples above. I have heavily contributed to three algorithms in ADAM: (1) Open Polynomial Dynamical Systems (oPDS), which allows the user to model open and deterministic biological systems, (2) Stochastic Discrete Dynamical Systems (SDDS), which can be used for modeling stochasticity and variability in closed systems, (3) Open Stochastic Discrete Dynamical Systems (oSDDS) which is an extended version of SDDS for modeling and simulating open and stochastic systems.

Chapter 3

A Mathematical Model of Denitrification in *Pseudomonas aeruginosa*

Oxygen (O_2) is necessary to sustain all life on Earth. In the aquatic system, oxygen dissolves in water, and assists living organisms such as fish to breathe. Hypoxia is the phenomenon of low dissolved oxygen. Although every animal has a different tolerance to hypoxia, it becomes dangerous when the level of dissolved oxygen is below $2mg\ O_2$ per Liter [14]. The main reasons for hypoxia in waters is an imbalance in the nitrogen cycle caused by increased human population, overfertilization, and industrial pollutants via drainage, from which nitrogen (N) and phosphorus (P) drain into the water [15]. Hypoxic (low-oxygen) areas are called dead zones, which Lake Erie has witnessed every summer for decades. Dead zones in Lake Erie lead to microbial production of the greenhouse gas nitrous oxide (N_2O), which plays an important role in ozone layer depletion and global warming. Thus, the microbial community of Lake Erie was scanned and Pseudomonas aeruginosa was chosen by my collaborator, molecular biologist Dr. George Bullerjahn at Bowling Green State University in Bowling Green, OH. This decision is based on the fact that *P. aeruginosa* is abundant in Lake Erie and is one of the most widely studied microbes in the literature. I constructed a metabolic network of *P. aeruginosa* in order to shed light on the environmental factors contributing to greenhouse gas accumulation in Lake Erie.

Pseudomonas aeruginosa is a rod-shaped, ubiquitous, and facultative bacterium well adapted to anaerobic conditions. Under anaerobic conditions and the presence of nitrate, *P. aeruginosa* can perform (complete) denitrification, a respiratory process of dissimilatory nitrate reduction to nitrogen gas via nitrogen oxides [16]. Complete denitrification consists of four sequential steps to reduce nitrate (NO_3) to dinitrogen (N_2) via nitrite (NO_2), nitric oxide (NO), and nitrous oxide (N_2O), and each step of the pathway is catalyzed by (denitrification) enzymes such as nitrate reductase (NAR), nitrite reductase (NIR), nitric oxide reductase (NOR), and nitrous oxide reductase (NOS). The bacterium uses these nitrogen oxides to grow under anaerobic conditions [17]. In this chapter, I will present a denitrification metabolic network of *P. aeruginosa* and its state and time discrete mathematical models. I will also analyze the steady states of the network by changing the concentration levels of environmental parameters such as oxygen (O_2), phosphate (PO_4), and nitrate (NO_3).

The regulatory network of *P. aeruginosa* is the third largest regulatory network among bacteria [18]. This project focuses on a snapshot of the entire regulatory network consisting of genes and enzymes playing an important role in the denitrification process. Figure 3.1 is a literature-based denitrification metabolic network in *Pseudomonas aeruginosa*. PO_4 down-regulates PhoPQ, and PhoPQ inhibits the expression of PmrA [18]. Low oxygen (O_2) tension, which is the major initial signal to turn on the denitrification pathway in *P. aeruginosa* [16], can be sensed by ANR (anaerobic regulation of arginine deiminase and nitrate reduction). Under anaerobic conditions, ANR activates NarXL in the presence of NO_3, and primarily promotes DNR (dissimilatory nitrate respiration regulator) transcription. The effect of ANR on DNR can be reduced by PmrA, and can be amplified by NarXL. NirQ, which can be activated by NarXL or DNR, regulates NIR and NOR coordinately to keep the level of NO low, because it is toxic not only for the environment but also for the microbe itself [17]. DNR is the NO sensor of the system. DNR induces the expression of all denitrification enzymes (NAR, NIR, NOR, NOS) in the presence of NO [19]. NarXL, which is a nitrate-responding regulatory system, directly activates NAR, and indirectly activates NIR and NOR via NirQ.

Figure 3.1: Denitrification Metabolic Network in *P. aeruginosa*

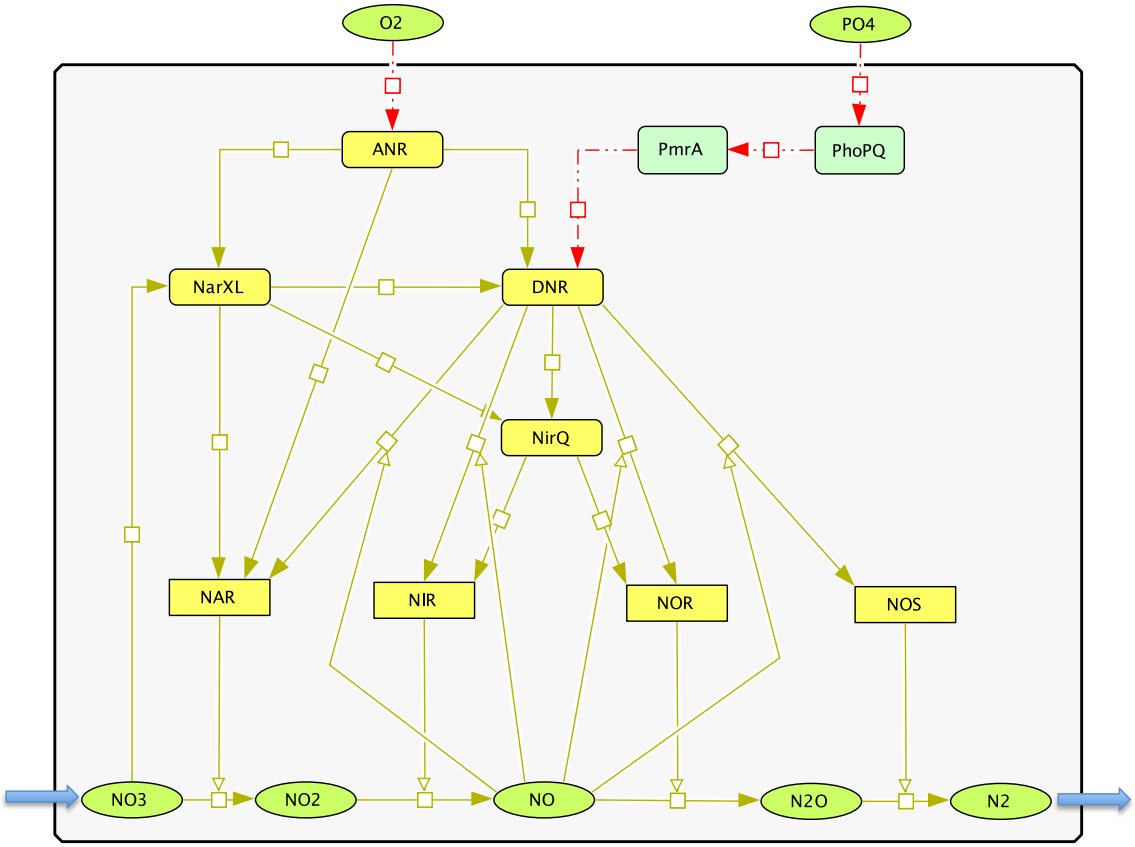


Table 3.1: Transition table for ANR

$O_2(t)$	$ANR(t+1)$
0	1
1	0

In Figure 3.1, green solid arrows indicate activation and red dashed arrows indicate inhibition. This denitrification metabolic network of *P. aeruginosa* has 14 variables (PhoPQ, PmrA, ANR, NarXL, DNR, NirQ, NAR, NIR, NOR, NOS, NO_2, NO, N_2O, N_2) and 3 external parameters (O_2, PO_4, NO_3). Here, PhoPQ, PmrA, ANR, NarXL, DNR, and NirQ are regulatory proteins; and NAR, NIR, NOR, and NOS are enzymes. This is an open system, meaning it exchanges molecules with the outside environment and respond to external stimuli [3]. Nitrate (NO_3) is the molecule entering the bacterium, and dinitrogen (N_2) is the molecule leaving the bacterium. The entire process occurs intracellularly.

3.1 Reduced Denitrification Metabolic Network

For model reduction, the determination of the variables influenced by external parameters was found using the following four steps:

Step 1. Based on the literature, ANR is activated under low O_2 tension, and activates DNR. Table 3.1 is the transition table of ANR, whose levels are determined by an external parameter O_2. Hence, it can be concluded that DNR is activated when the concentration level of O_2 is low.

Step 2. Table 3.2 represents the transition table for NarXL. Note that NarXL is activated only when the concentration level of nitrate (NO_3) is high and ANR is expressed. Therefore, NarXL

Table 3.2: Transition table for NarXL

$NO_3(t)$	$ANR(t)$	$NarXL(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1

Table 3.3: Transition table for PhoPQ

$PO_4(t)$	$PhoPQ(t+1)$
0	1
1	0

can be activated by NO_3 and O_2 since the activation of ANR depends upon the low oxygen (O_2) tension.

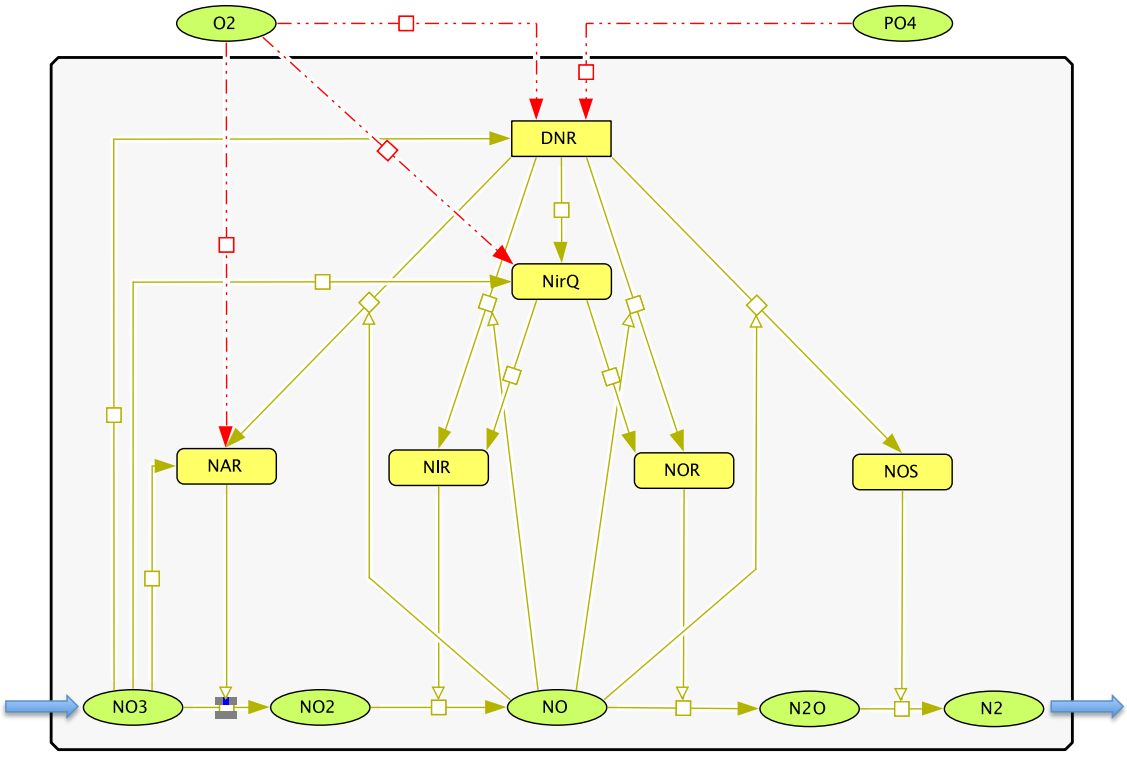
Step 3. Because PmrA is inhibited by PhoPQ, which is activated when the concentration level of phosphate (PO_4) is low (see Table 3.3), it can be concluded that the activation of PmrA depends upon the concentration level of phosphate (PO_4) in the environment.

Step 4. The transition table for PmrA (Table 3.4) indicates that PmrA is inhibited by PhoPQ, which is inhibited under high PO_4 conditions. In other words, the activation of PmrA is indirectly related to the concentration level of PO_4. Since DNR is inhibited by PmrA, DNR is (indirectly) inhibited when PO_4 is abundant in the environment.

Table 3.4: Transition table for PmrA

$PhoPQ(t)$	$PmrA(t+1)$
0	1
1	0

Figure 3.2: Reduced Denitrification Metabolic Network in *P. aeruginosa*



Since ANR, NarXL, PmrA, and PhoPQ can easily be determined by external parameters, they do not need to be included in the system. Figure 3.2 depicts the *reduced* metabolic network of *P. aeruginosa* with 10 variables (DNR, NirQ, NAR, NIR, NOR, NOS, NO_2, NO, N_2O, N_2) preserving the characteristics of the system. All variables in the model are Boolean (0 = low/inactive, 1 = high/active) including external parameters except DNR. Since the activation of DNR can be reduced by PO_4 (via PmrA) and can be amplified by NO_3 (via NarXL), DNR can take on 3 values (0 = inactive, 1 = active, 2 = super-active). To model the network (Figure 3.2), I first utilize a deterministic framework, Polynomial Dynamical Systems (PDS), and then a stochastic framework, Stochastic Discrete Dynamical Systems (SDDS), with a *synchronous* update schedule, meaning all variables in the system are updated simultaneously [20]. Both models are the first attempt to model the denitrification metabolic network in *P. aeruginosa* in the literature.

3.2 Polynomial Dynamical System Model

In the PDS model, there are ($n =$)10 variables, each variable can take up to ($p =$)3 different values. Note that $\mathbb{F} = F_3$. Let the variables be labeled as follows:

$$
\begin{aligned}
x_1 &= DNR & x_6 &= NOS \\
x_2 &= NirQ & x_7 &= NO_2 \\
x_3 &= NAR & x_8 &= NO \\
x_4 &= NIR & x_9 &= N_2O \\
x_5 &= NOR & x_{10} &= N_2
\end{aligned}
$$

The transition table of each variable is constructed and then converted to a polynomial function using the Equation 2.1. For instance, the update function of NOS (x_6) can be obtained using its transition table, Table 3.5, which indicates NOS is activated when NO (x_8) is present, and DNR

Table 3.5: Transition table for NOS

$x_1(t)$	$x_8(t)$	$x_6(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	0
2	1	1

(x_1) is expressed. Then, the (polynomial) update function of NOS (x_6) is

$$f_6(x) = \sum_{(c_1,c_8)\in\mathbb{F}^2} h(c_1,c_8) \prod_{j\in\{1,8\}} (1-(x_j-c_j)^{3-1}) \pmod 3$$

$$\begin{aligned} &= 0*((1-(x_1-0)^2)*(1-(x_8-0)^2)) + 0*((1-(x_1-0)^2)* \\ &\quad (1-(x_8-1)^2)) + 0*((1-(x_1-1)^2)*(1-(x_8-0)^2)) + 1* \\ &\quad ((1-(x_1-1)^2)*(1-(x_8-1)^2)) + 0*((1-(x_1-2)^2)* \\ &\quad (1-(x_8-0)^2)) + 1*((1-(x_1-2)^2)*(1-(x_8-1)^2)) \pmod 3 \\ &= -x_1^2*x_8^2 - x_1^2*x_8 \end{aligned}$$

Other update functions as polynomials have been constructed in the same manner. The literature-based transition table for each variable in the system is provided in Appendix A. There are no transition tables for external parameters (O_2, PO_4, NO_3) since my interest lied in perturbation of these external parameters and their effect on the long-term behavior of the system. The update functions for all variables (DNR, NirQ, NAR, NIR, NOR, NOS, NO_2, NO, N_2O, N_2) are:

$$\begin{aligned} f_1 = &-O_2^2*NO_3^2*PO_4^2 - O_2^2*NO_3^2*PO_4 - O_2^2*NO_3*PO_4 + O_2^2*PO_4^2 + NO_3^2*PO_4^2 - \\ &O_2^2*NO_3 + NO_3^2*PO_4 - O_2^2 + NO_3*PO_4 - PO_4^2 + NO_3 + 1 \end{aligned}$$

$$f_2 = O_2^2 * NO_3 * x_1^2 - O_2 * NO_3^2 * x_1^2 + O_2^2 * NO_3^2 + O_2^2 * x_1^2 + O_2 * NO_3 * x_1^2 - NO_3^2 * x_1^2 + O_2^2 * NO_3 - O_2 * x_1^2 - NO_3^2 + x_1^2 - NO_3$$

$$f_3 = -O_2^2 * NO_3^2 * x_1^2 * x_8^2 - O_2^2 * NO_3^2 * x_1^2 * x_8 + O_2^2 * NO_3^2 * x_8^2 + O_2^2 * x_1^2 * x_8^2 + NO_3^2 * x_1^2 * x_8^2 - O_2^2 * NO_3^2 * x_8 + O_2^2 * x_1^2 * x_8 + NO_3^2 * x_1^2 * x_8 + O_2^2 * NO_3 * x_8^2 + O_2^2 * NO_3^2 - O_2^2 * NO_3 * x_8 - NO_3^2 * x_8^2 - x_1^2 * x_8^2 + O_2^2 * NO_3 + NO_3^2 * x_8 - x_1^2 * x_8 - NO_3 * x_8^2 - NO_3^2 + NO_3 * x_8 - NO_3$$

$$f_4 = x_1^2 * x_2^2 * x_8^2 + x_1^2 * x_2^2 * x_8 - x_1^2 * x_8^2 - x_2^2 * x_8^2 - x_1^2 * x_8 + x_2^2 * x_8 - x_2 * x_8^2 - x_2^2 + x_2 * x_8 - x_2$$

$$f_5 = x_1^2 * x_2^2 * x_8^2 + x_1^2 * x_2^2 * x_8 - x_1^2 * x_8^2 - x_2^2 * x_8^2 - x_1^2 * x_8 + x_2^2 * x_8 - x_2 * x_8^2 - x_2^2 + x_2 * x_8 - x_2$$

$$f_6 = -x_1^2 * x_8^2 - x_1^2 * x_8$$

$$f_7 = NO_3^2 * x_3^2 + NO_3^2 * x_3 + NO_3 * x_3^2 + NO_3 * x_3$$

$$f_8 = x_4^2 * x_7^2 + x_4^2 * x_7 + x_4 * x_7^2 + x_4 * x_7$$

$$f_9 = x_5^2 * x_8^2 + x_5^2 * x_8 + x_5 * x_8^2 + x_5 * x_8$$

$$f_{10} = x_6^2 * x_9^2 + x_6^2 * x_9 + x_6 * x_9^2 + x_6 * x_9$$

Thus, the PDS corresponding to this network is

$$f = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}) : \mathbb{F}^{10} \to \mathbb{F}^{10}.$$

With 1 simulation and 25 time steps, Figure 3.3 shows the long-term behavior of DNR, NOS, N_2O, and N_2 under the perfect condition for denitrification; low O_2, low PO_4, and high NO_3. Here, the initial state is (0, 0, 0, 0, 0, 0, 0, 0, 0, 0); low oxygen in the environment is not yet sensed. In these conditions, DNR is expected to be highly expressed, and then NOS is supposed to be induced (by DNR). Therefore, N_2O can be converted to N_2, i.e. the bacterium performs (complete) denitrification. Figure 3.4 illustrates the trajectory starting with (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) and ending at the steady state of the system, (2, 1, 1, 1, 1, 1, 1, 1, 1, 1).

Recall that microbial networks are highly stochastic [11]; there is some probability that the process may not occur due to stochasticity, and that there may be more than one next state for a given state. In the next section, this stochasticity will be introduced to the model at the update function level.

Figure 3.3: Cell population dynamics of *P. aeruginosa* during denitrification in PDS model

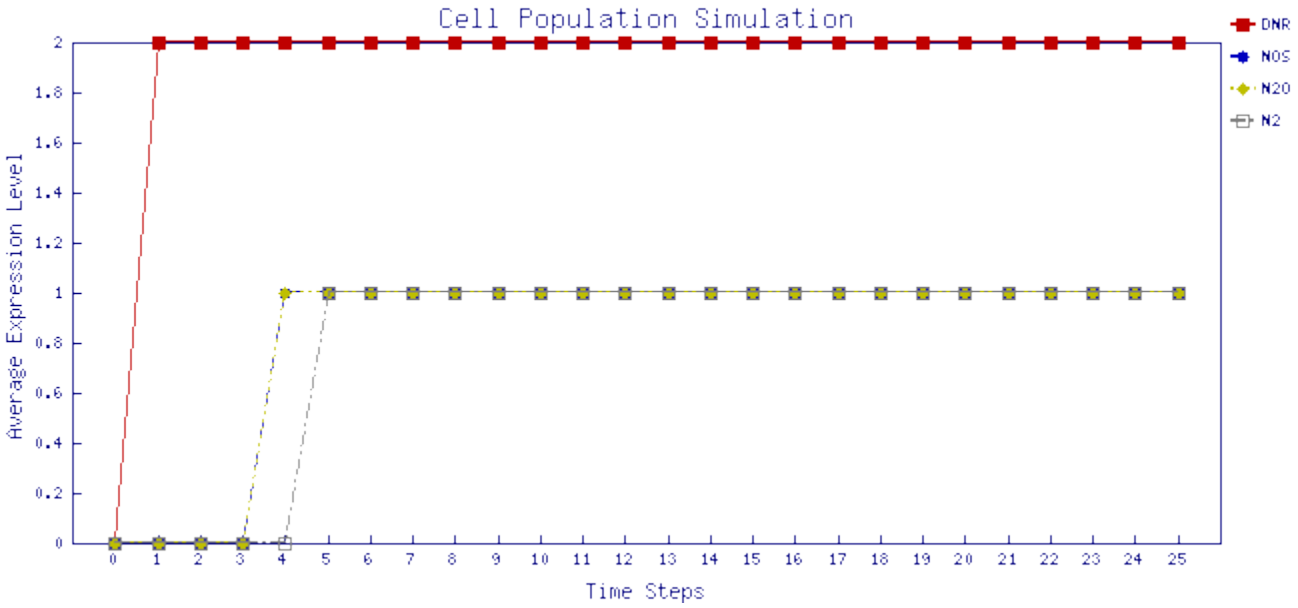
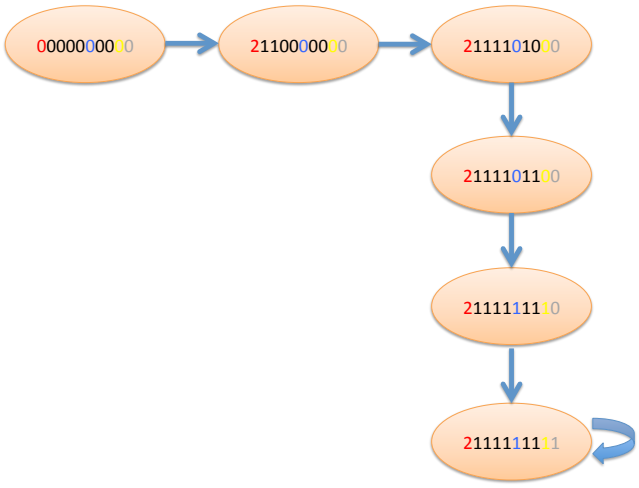


Figure 3.4: The trajectory with the initial state (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) in PDS model



3.3 Stochastic Discrete Dynamical System Model

In the SDDS model, there are (n =)10 variables, each variable can take up to (p =)3 different values. Suppose the system is randomly updated, i.e. activation and degradation propensities are all 0.5. Then, the SDDS model of this system is

$$\{f_i, 0.5, 0.5\}_{i=1}^{10}$$

where f_i's are an update function of x_i provided in the PDS model in Section 3.2.

With 100 simulations, 25 time steps, and (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) as an initial state, Figure 3.5 shows the long-term behavior of DNR, NOS, N_2O, and N_2 under the perfect condition for denitrification: low O_2, low PO_4, and high NO_3. Figure 3.6 illustrates the trajectory starting with (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) and ending at the steady state of the system, (2, 1, 1, 1, 1, 1, 1, 1, 1, 1). The numbers adjacent to the edges are the transition probabilities.

Model construction and analysis are done in Analysis of Dynamic Algebraic Models (ADAM), available at http://dvd.vbi.vt.edu/adam.html. The models can be tested choosing open Polynomial Dynamical System (oPDS), and open Stochastic Discrete Dynamical System (oSDDS) as model types in ADAM.

While microbial networks are stochastic, introducing stochasticity into the system would not make a big difference to the long-term behavior of the system. Based on the sensitivity analysis of propensity parameters in the SDDS model, the system does not have an oscillatory behavior due to the fact that there is not any negative feedback loop in the network and there is only one steady state in the system under any environmental conditions. Therefore, the PDS model will be used as a mathematical model of the denitrification metabolic network in *Pseudomonas aeruginosa* until a negative feedback loop is needed to be introduced into the system.

Figure 3.5: Cell population dynamics of *P. aeruginosa* during denitrification in SDDS model

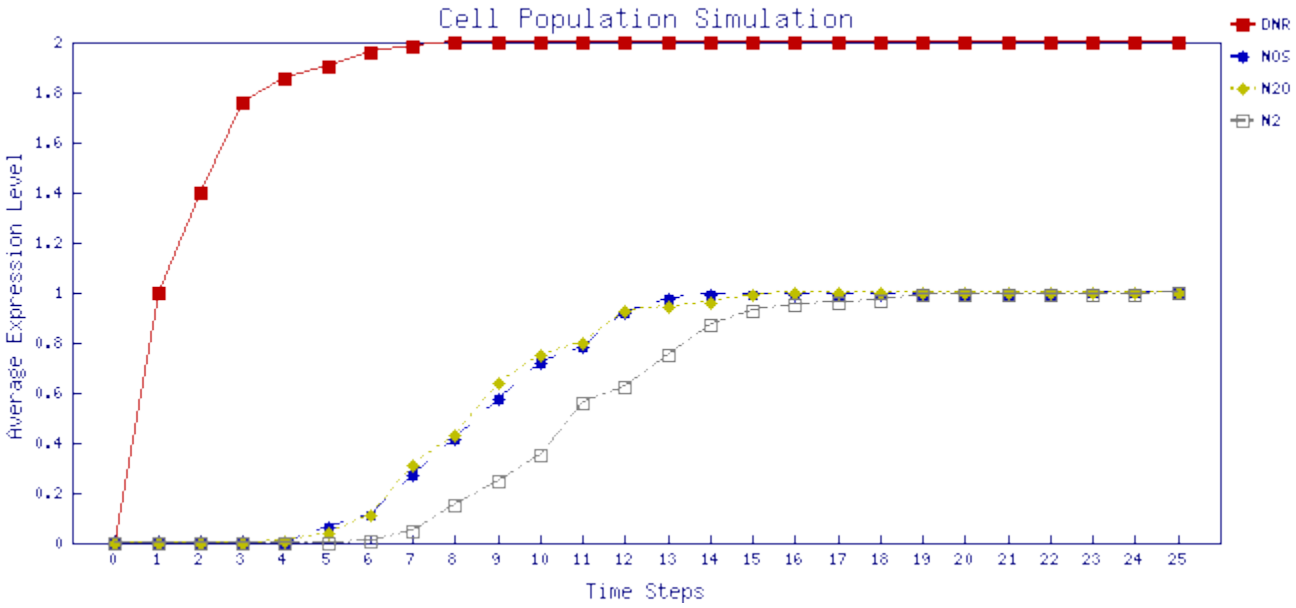


Figure 3.6: Part of the state space with the initial state (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) in SDDS model

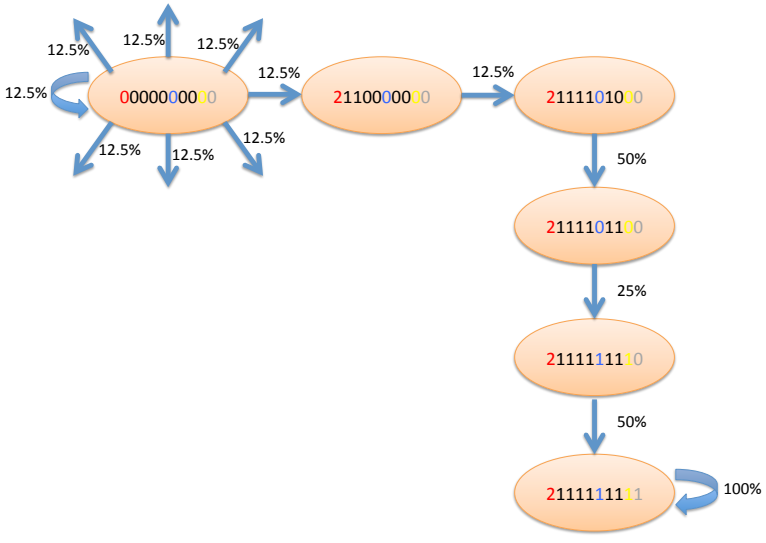


Table 3.6: Steady states of the system under all possible environmental conditions

O_2	PO_4	NO_3	DNR	$NirQ$	NAR	NIR	NOR	NOS	NO_2	NO	N_2O	N_2
0	0	0	1	1	0	1	1	0	0	0	0	0
0	0	1	2	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0

3.4 Analysis

It is worth keeping in mind that open systems eventually reach a (homeostatic) steady state and are in balance with their environment [3]. Table 3.6 indicates the steady state analysis of the system under different environmental conditions. All results except the blue ones confirm pertinent biological literature:

- If the concentration level of all external parameters are low, then, DNR is expected to be activated, so is NirQ. NirQ activates NIR and NOR in order to keep the NO level low [21]. The model provides the steady state of the system as (1, 1, 0, 1, 1, 0, 0, 0, 0, 0), the state where high nitrate tension is not sensed; NAR is not induced, so there is no reduction of NO_3 to N_2 via nitrogen oxides.

- If the concentration levels of O_2 and PO_4 are low, and NO_3 is high, then it is a perfect con-

Table 3.7: Biological interpretation of the steady states of the system under all possible environmental conditions

O_2	PO_4	NO_3	BIOLOGICAL INTERPRETATION
0	0	0	no denitrification - high NO_3 tension is not sensed
0	0	1	high denitrification performance
0	1	0	no denitrification
0	1	1	low denitrification performance
1	0	0	no denitrification - low O_2 tension is not sensed
1	0	1	no denitrification - low O_2 tension is not sensed
1	1	0	no denitrification - low O_2 tension is not sensed
1	1	1	no denitrification - low O_2 tension is not sensed

dition for complete denitrification to N_2; therefore, all variables in the network are expected to be active/high. The model provides the steady state of the system as (2, 1, 1, 1, 1, 1, 1, 1, 1, 1), the state where the bacterium performs denitrification. DNR is super-active, denitrification enzymes are active, and NO_3 is converted to N_2 since DNR is (highly) activated under anaerobic conditions as well as other regulatory proteins and denitrification enzymes [17].

- If the concentration levels of O_2 and NO_3 are low, and PO_4 is high, then the model suggests that (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) is the steady state of the system. This case should be studied for confirmation.

- If the concentration level of O_2 is low, PO_4 and NO_3 are high, then the model suggests that (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) is the steady state of the system. This case is currently under study. However, the preliminary lab work did show a modest increase in N_2O production with a high PO_4 level. This is an example of how PO_4 can influence the expression of variables distant from PO_4 acquisition.

- If the concentration level of O_2 is high, then regardless of the concentration level of other external parameters (PO_4 or NO_3), *Pseudomonas aeruginosa* cannot perform denitrification due to the fact that DNR, the main regulator of the system, cannot be activated in the presence of oxygen [16]. The variables in the system are expected to be inactive/low, and the nitrogen oxides are expected not to be reduced. The model provides the steady state of the system as (0, 0, 0, 0, 0, 0, 0, 0, 0, 0), the state where low oxygen tension is not sensed.

Using the constructed models and *in-silico* experiments, I knock down DNR in order to see how the system is affected during denitrification when the main regulator of the system is off. For this, the update function of DNR (x_1) is set to zero and the external parameters are set as follows: $O_2 = 0$, $PO_4 = 1$, and $NO_3 = 1$. The simulations show that the steady state of the system with knocked down DNR is (0, 1, 1, 1, 1, 0, 1, 1, 1, 0), meaning the only denitrification enzyme completely depending upon active DNR is NOS (x_6). Therefore, the last step of denitrification, the reduction of N_2O to N_2, can not be processed to a full extent. In light of this *in-silico* experiment, I hypothesize that there are 2 reasons for the greenhouse gas N_2O accumulation due to inactive NOS:

Hypothesis 1. PO_4 inhibits the induction of NOS by DNR.

Recall that either NarXL or NirQ can induce all denitrification enzymes except NOS. The only variable that activates NOS is DNR (Figure 3.1). Hence, I propose PO_4 may prevent NOS to be induced by DNR.

Hypothesis 2. PO_4 inhibits the activation of DNR by ANR.

Since DNR is the only known regulatory protein activating NOS, inactivation of DNR can induce inactive NOS, resulting in N_2O accumulation. DNR is activated by only ANR (Figure 3.1),which leads to the conclusion that PO_4 may prevent DNR to be activated by ANR.

These hypotheses (the effect of PO_4 into the network) are currently being tested by my collaborator, Dr. George Bullerjahn's research group, using gas chromatography (GC) and reverse transcriptase quantitative PCR (qRT-PCR). GC is used for measuring the output of N_2O from cultures grown under increasing PO_4 conditions. PO_4 is a medium component, and it is tested whether there is more N_2O in high PO_4 cultures compared to low PO_4 cultures, a control culture lacking PO_4, and another uninoculated high PO_4 control culture during denitrification. qRT-PCR, on the other hand, is utilized to detect levels of mRNA from target genes such as denitrification genes and *dnr* gene encoding the regulatory protein, DNR. RNA is isolated from denitrifying cultures, and the rate at which the genes are amplified (measured by the incorporation of a fluorescent tracer molecule) is a function of mRNA abundance.

Chapter 4

Conclusion

Mathematical modeling, the heart of systems biology, aids scientists to systematically gather information about the system of interest, grasp the key interactions and features of the system, design computer programs to simulate the system, canalize the flow of biological experiments, and detect incorrect results from previous studies on the system of interest (if any). Despite the many varieties of model strategies in the literature, this thesis focuses on multi-state and time discrete dynamical systems: Polynomial Dynamical Systems (PDS) and Stochastic Discrete Dynamical Systems (SDDS). A PDS, a deterministic mathematical framework, uses polynomial functions over a finite field in order for programmability and analyzability with polynomial time complexity. On the other hand, an SDDS, a stochastic mathematical framework, uses activation and degradation propensities in order to model stochasticity at the biological function level.

This thesis describes a denitrification metabolic network of *Pseudomonas aeruginosa*, and its *initial* PDS and SDDS models in order to shed light on anaerobic production of the greenhouse gas N_2O in Lake Erie, OH. Analysis and simulations of the models confirm currently available information in pertinent literature. *In-silico* experiments provide two hypotheses for N_2O accumulation, which are currently being tested at Dr. George Bullerjahn's lab: High PO_4 concentrations may prevent the activation of DNR by ANR, or the induction of NOS by DNR. Preliminary results

indicate that N_2O production is noticeably increased in high PO_4 cultures.

Our goals for the future include:

- model validation for the presented denitrification metabolic network in *P. aeruginosa* using the experimental methods described in [22] for *nosZ*, and the primer sets described in [23] for *narG* and *nirS*,
- construction of an agent-based model of population dynamics in *P. aeruginosa* during denitrification in order to understand how a group of the bacteria affects its denitrification performance,
- investigation of the relationship between the denitrification metabolic networks in P. aeruginosa and in other microbes in Lake Erie in the interest of a common metabolic consensus network as a computational model.

Bibliography

[1] J. Bailey, "Mathematical modeling and analysis in biochemical engineering: Past accomplishments and future opportunities," *Biotechnology Progress*, vol. 14, no. 1, pp. 8–20, 1998.

[2] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall / CRC, 2006.

[3] B. Palsson, *Systrems Biology: Simulation of Dynamic Network States*. Cambridge University Press, 2011.

[4] R. M. Thrall, J. A. Mortimer, K. R. Rebman, and R. F. Baum, *Some Mathematical Models in Biology*, revised ed. The University of Michigan National Technical Information Service, 1967.

[5] D. J. Wilkinson, *Stochastic Modelling for Systems Biology*. Chapman & Hall / CRC, 2012.

[6] A. Veliz-Cuba, A. S. Jarrah, and R. Laubenbacher, "Polynomial algebra of discrete models in systems biology," *Bioinformatics*, vol. 26, no. 13, pp. 1637–43, 2010.

[7] H. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, 1998.

[8] A. Veliz-Cuba, "An algebraic approach to reverse engineering finite dynamical systems arising from biology," *SIAM J. Applied Dynamical Systems*, vol. 11, no. 1, pp. 31–48, 2012.

[9] R. Laubenbacher and B. Stigler, "A computational algebra approach to the reverse engineering of gene regulatory networks," *J Theor Biol*, vol. 229, no. 4, pp. 523–37, 2004.

[10] R. Lidl, H. Niederreiter, and P. Cohn, *Finite fields*. Cambridge University Press, 1997, vol. 20.

[11] A. Raj and A. van Oudenaarden, "Stochastic gene expression and its consequences," *Cell*, vol. 135, no. 2, pp. 216–226, 2008.

[12] D. Murrugarra, A. Veliz-Cuba, B. Aguilar, S. Arat, and R. Laubenbacher, "Modeling stochasticity and variability in gene regulatory networks," *EURASIP J Bioinform Syst Biol*, vol. 2012, no. 1, p. 5, 2012.

[13] F. Hinkelmann, M. Brandon, B. Guang, R. McNeill, G. Blekherman, A. Veliz-Cuba, and R. Laubenbacher, "Adam: Analysis of discrete models of biological systems using computer algebra," *BMC Bioinformatics*, vol. 12, p. 295, 2011.

[14] R. J. Diaz, "Overview of hypoxia around the world," *Journal of Environmental Quality*, vol. 30, no. 2, pp. 275–281, 2001.

[15] C. L. Dybas, "Dead zones spreading in world oceans," *Bioscience*, vol. 55, no. 7, pp. 552–557, 2005.

[16] W. G. Zumft, "Cell biology and molecular basis of denitrification," *Microbiol Mol Biol Rev*, vol. 61, no. 4, pp. 533–616, 1997.

[17] H. Arai, "Regulation and function of versatile aerobic and anaerobic respiratory metabolism in pseudomonas aeruginosa," *Front Microbiol*, vol. 2, p. 103, 2011.

[18] E. Galan-Vasquez, B. Luna, and A. Martinez-Antonio, "The regulatory network of pseudomonas aeruginosa," *Microb Inform Exp*, vol. 1, no. 1, p. 3, 2011.

[19] G. Giardina, S. Rinaldo, K. A. Johnson, A. Di Matteo, M. Brunori, and F. Cutruzzola, "No sensing in pseudomonas aeruginosa: structure of the transcriptional regulator dnr," *J Mol Biol*, vol. 378, no. 5, pp. 1002–15, 2008.

[20] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *J Comput Biol*, vol. 9, no. 1, pp. 67–103, 2002.

[21] H. Arai, T. Kodama, and Y. Igarashi, "Effect of nitrogen oxides on expression of the nir and nor genes for denitrification in pseudomonas aeruginosa," *FEMS Microbiol Lett*, vol. 170, pp. 19–24, 1999.

[22] S. Henry, D. Bru, B. Stres, S. Hallet, and L. Philippot, "Quantitative detection of the nosz gene, encoding nitrous oxide reductase, and comparison of the abundances of 16s rrna, narg, nirk, and nosz genes in soils," *Appl Environ Microbiology*, vol. 72, no. 8, p. 51815189, 2006.

[23] E. Kandeler, D. K., D. Tscherko, D. Bru, and L. Philippot, "Abundance of narg, nirs, nirk, and nosz genes of denitrifying bacteria during primary successions of a glacier foreland," *Applied and Environmental Microbiology*, vol. 72, no. 9, p. 59575962, 2006.

Appendix A

Transition Tables for All Variables in the Model

Table A.1: Transition table for DNR

$O_2(t)$	$PO_4(t)$	$NO_3(t)$	$x_1(t+1)$
0	0	0	1
0	0	1	2
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Table A.2: Transition table for NirQ

$O_2(t)$	$NO_3(t)$	$x_1(t)$	$x_2(t+1)$
0	0	0	0
0	0	1	1
0	0	2	1
0	1	0	1
0	1	1	1
0	1	2	1
1	0	0	0
1	0	1	1
1	0	2	1
1	1	0	0
1	1	1	1
1	1	2	1

Table A.3: Transition table for NAR

$O_2(t)$	$NO_3(t)$	$x_1(t)$	$x_8(t)$	$x_3(t+1)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	0	2	0	0
0	0	2	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
0	1	2	0	1
0	1	2	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	0	2	0	0
1	0	2	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0
1	1	2	0	0
1	1	2	1	0

Table A.4: Transition table for NIR

$x_1(t)$	$x_2(t)$	$x_8(t)$	$x_4(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
2	0	0	0
2	0	1	1
2	1	0	1
2	1	1	1

Table A.5: Transition table for NOR

$x_1(t)$	$x_2(t)$	$x_8(t)$	$x_5(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
2	0	0	0
2	0	1	1
2	1	0	1
2	1	1	1

Table A.6: Transition table for NOS

$x_1(t)$	$x_8(t)$	$x_6(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	0
2	1	1

Table A.7: Transition table for NO_2

$NO_3(t)$	$x_3(t)$	$x_7(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1

Table A.8: Transition table for NO

$x_4(t)$	$x_7(t)$	$x_8(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1

Table A.9: Transition table for N_2O

$x_5(t)$	$x_8(t)$	$x_9(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1

Table A.10: Transition table for N_2

$x_6(t)$	$x_9(t)$	$x_{10}(t+1)$
0	0	0
0	1	0
1	0	0
1	1	1