UNIVERSITY OF SOUTHAMPTON

Mathematical Model of Competence Regulation Circuit

by

An Nguyen

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Electronics and Computer Science

December 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by An Nguyen

Gene expression regulatory networks are molecular networks which describe interactions among gene products in terms of biochemical reactions. This helps us understand the molecular mechanisms underlying important biological processes as well as cell functioning as a whole. For instance, the phenomenon of bacterial competence, whereby a bacterium enters a transiently differentiated state, incorporating DNA fragments from its environment into its genome, has been studied with the help of such gene regulatory circuits (Süel et al., 2006; Maamar and Dubnau, 2005). As a result, a genetic circuit has been taken into account in order to describe the transition from a vegetative state to a transient state of competence and vice versa. In this work, we are going to study a genetic circuit presented by Süel et al. (2007) to describe this dynamical behaviour. The authors introduce model reduction techniques to study the behaviour of stochastic chemical system of X species by means of an adiabatic two dimensional model. While the adiabatic model helps us understand about the dynamics near the steady state, it gives an incorrect description of the time-scales of the competent state. For this reason, it is necessary to build up a model which better describes the system realistically. In the thesis, I propose an approximate two-dimensional model of the full high-dimensional system and from that, the dynamics of the system can be simulated more accurately compared to that of Süel et al. (2007). I then show how to put the noise back into the approximate model to be able come up with a stochastic model which can mathematically describe the dynamical behaviour of the original high dimensional system. I also found out that the evolution of the system is not well approximated by a Langevin process. This leads to a gap between the real behaviour which is described by Gillespie's stochastic simulation and the Langevin approximation. To overcome this, I have fixed the stochastic Langevin model by incorporating empirically tunable noise into the model so as to obtain a similar behaviour as observed in the original system. I also introduce the chemical Fokker-Planck equation aimed to estimate the probability density function of species concentrations which are involved in the biochemical system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, I will be presenting some basic background on the methods used in systems biology to understand the dynamical behaviour of gene regulation networks within biological systems. I introduce mathematical tools which are necessary for understanding the system of concern, these include the Chemical Master Equation (CME), Gillespie's algorithm, the Chemical Langevin Equation (CLE), the Reaction Rate Equation (RRE), and the Fokker-Planck equation. I will then be talking about the motivations of thesis in which I will point out the critical reasons why we need to conduct the research as well as the main contributions of the thesis. The structure of the thesis is then outlined in more details.

1.1 Background

In order to understand the problem we are going to discuss in the thesis, it is necessary to understand some basics of systems biology which will be detailed in the following section, as a large picture of the research area. We will then focus on a particular simple system which has been widely studied to describe a phenomenon called competence in bacteria *Bacillus subtilis*. I also introduce a series of mathematical tools which have been used to describe the behaviour of the system.

1.1.1 Systems Biology

Systems biology is the study of systems of biological components such as molecules, cells, organisms or species, focused on the complex interactions within the biological system. This means the aim of this study is to examine the structure and complex interaction among those biological components as a system, rather than the activity of independent parts of a cell or organism (Kitano, 2002). Over the last couple of decades, many researches in systems biology have also been conducted to better understand the

behaviour of complicated biological systems at cellular level. The most challenging problem to understanding this is to deal with a huge number of species as well as chemical reactions involved. Consequently, this has led to an increasing attempt to describe the complex biological processes using techniques such as metabolic pathways, gene regulation networks and cell signaling pathways. However, in this thesis, we will focus on gene regulation networks (GRNs) to study the regulation mechanism in gene expression, which basically includes transcription and translation processes.

In living cells, regulation of gene expression is a fundamental mechanism of development of adaptation to a variable environment. On the other hand, control of expression is vital for cells to produce gene products when needed so that this gives cells the flexibility to respond to external signals or damage to the cells (López-Maury et al., 2008). This process includes changes of all structure and function of cells ranging from transcription of DNA into RNA to translation of mRNA into protein, and, finally, post-translational modification of protein into its mature, functional form (Phillips, 2008).

Transcription regulation is the way a cell controls the speed and quantity of production of functional proteins to respond to the needs of an organism. This mechanism is regulated by transcription factor proteins that modulate the efficiency of mRNA transcription generation from which the proteins are transcribed. The transcription factor proteins can either facilitate (activator) the expression of a specific gene in order to stimulate producing its own proteins or inactivate (repressor) other genes to prevent them from overproducing products which may not be necessary for the process (White, 2001).

In this thesis, we are interested in the regulation mechanism for a biological phenomenon called competence in bacteria. Competence is a physiological state which enables cells to bind and internalize DNA from its environment (Dubnau, 1991; Turgay et al., 1998). On the other hand, competence may occur either under natural conditions such as heat, nutrition limitation, etc., or in the laboratory where cells are made transiently permeable to DNA (Hatami et al., 2004). In bacteria, the DNA uptake may help them respond to environmental stresses as well as be able to survive under adverse conditions. One of bacterial species which has been considered to have a high level of competence is *Bacillus subtilis*. In this bacteria, competence usually occurs at a specific stage of growth where the number of cells and the rates of population increase doubles with each consecutive time period just before cells enter the stationary phase. At this state, the number of cells remains at a constant value. As a result, a cell may make a transition from a static or vegetative state to the state of competence in which the surrounding DNA could be absorbed. In experiment, by using fluorescent markers in cell-sorting (Shimomura et al., 1962; Herzenberg et al., 2002; Lippincott-Schwartz and Patterson, 2003) and time-lapse recording technologies (Hinchcliffe, 2005), researchers have found that single cell fluorescent images reveal bimodal cell populations of a key transcription factor protein that is used to track the competent state, ComK. As a result, cells can express ComK at low (vegetative state) or high (competent state) level. An example of this switching-like

behaviour can be described as bistability in mathematical models (Dubnau and Losick, 2006; Veening et al., 2008), which have been applied in synthetic systems with positive feedback switches (Gardner et al., 2000; Cheemeng et al., 2009).

The critical role of noise in making the transition between low and high expression levels was also reported in two studies. Elowitz and co-workers (Elowitz et al., 2002) constructed strains of *Escherichia coli* for detecting noise and showed that intrinsic noise increases as the transcription rate decreases. Based on this approach, Maamar and co-workers (Maamar et al., 2007) showed that intrinsic noise in ComK expression selects cells for competence. In particular, reducing the noise by increasing the transcription rate and decreasing the translation rate while keeping the protein concentration constant, results in fewer transition to the competent state. Their findings are also observed in another report that demonstrated the significant variation in basal expression rate of ComK (Leisner et al., 2007a). As a result, noise is a competence trigger and should be incorporated into genetic circuits for describing gene regulation. Recently, some noise-induced genetic circuits have been constructed to model the bimodal cells population of ComK in competence (Süel et al., 2006; Maamar et al., 2007; Dandach and Khammash, 2010). Bimodal probability distributions can be generated by an underlying stochastic dynamical system whose deterministic state is bistable. It is also demonstrated that the positive feedback provided by ComK proteins activating its own transcription can generate switching behaviour, whose noise-induced activation yields bimodal distributions (Maamar and Dubnau, 2005; Leisner et al., 2007b, 2009). An alternative model has been proposed by Süel et al. (2006, 2007) which includes a slower, negative influence on ComK levels via the expression of the *comK* gene. This leads to an excitable system, whose high expression (competent) state is not stable, but undergoes slow decay back to the low-expression (vegetative) state. The accumulation of ComK at this slowly decaying high expression accounts for the second mode of the bimodal distribution in the model.

In order to mathematically describe the relationships of chemical variables in genetic circuits, we use mathematical models including deterministic and stochastic models. The deterministic models are based on a set of differential equations describing the time-evolution of system state given an initial state, whereas the stochastic models include random factors in the underlying processes. The stochastic descriptions are necessary for the modelling the true dynamical behaviour of the system as the occurrence of small number of species makes the deterministic model inaccurate. To the system perspective, the stochastic models are needed when biologically observed phenomena are driven by stochastic fluctuations (for example, switching between the vegetative state and competent state). The stochastic models can be described by using stochastic simulation algorithm (SSA) which is set of numerical techniques for numerically simulating the time evolution of the given chemical system. Since Gillespie (Gillespie, 1977) first introduced the numerical stochastic formulation of chemical kinetics, this method has been applied to several well-known model chemical systems such as the Michaelis-Menten model, the

Schlögl model, the Lotka-Volterra model (Facoltá et al., 2007). Since all the results mentioned in the thesis are collected and computed from the simulation data, the starting point for all simulations is based on the Gillespie's algorithm. A description of Gillespie's stochastic algorithm will be detailed in the following section.

1.1.2 Gillespie's Stochastic Algorithm

Firstly, let's assume that the dynamcial state of our system is denoted as $\mathbf{X}(t) \equiv (X_1(t), X_2(t), \dots X_N(t))^\intercal$, where $X_i(t)$ is the number of molecules of each species $S_i (i = 1, ..., N)$ which interact to each other inside volume Ω of the system at time t. For each reaction $R_j (j = 1, ..., M)$, we define vector $\boldsymbol{\nu}_j \equiv (\nu_{1j}, ..., \nu_{Nj})$ called the stoichiometry with element ν_{ij} being the change in the amount of molecular number S_i. The probability given $\mathbf{X}(t)$, that a reaction R_j will occur in the time interval $[t, t + dt)$ causing a jump to state $\mathbf{X} + \boldsymbol{\nu}_j$ is given by $a_j(\mathbf{X})dt$, where $a_j(\mathbf{X})$ is the propensity function. For example, given the simple reaction below :

$$\begin{aligned} \mathrm{S}_1 + \mathrm{S}_2 &\xrightarrow{c_1} 2\mathrm{S}_1 \\ 2\mathrm{S}_1 &\xrightarrow{c_2} \mathrm{S}_1 + \mathrm{S}_2 \end{aligned} \tag{1.1}$$

where c_1, c_2 are stochastic rate constants. The probability that a pair of S_1 and S_2 is randomly chosen will react in the next infinitesimal time dt is given by $c_1 dt$. Thus, the probability that X_1 of the S_1 molecules will react with X_2 of the S_2 molecules in the next dt is $c_1 X_1 X_2 dt$, that implies that the propensity function for this reaction is $a_1(\mathbf{X}) = c_1 X_1 X_2$. For the reverse reaction, there are $\frac{X_1(X_1-1)}{2}$ possibilities of pairs $S_1 - S_1$ to react; therefore, the propensity function is $a_2(\mathbf{X}) = \frac{c_2 X_1(X_1-1)}{2}$.

In fact, $\mathbf{X}(t)$ is a discrete variable with probability $P(\mathbf{X}, t|\mathbf{X}_0, t_0)$ that the chemical system will be in state $\mathbf{X}$ at time t given $\mathbf{X}(t_0) = \mathbf{X}_0$. We are now interested in the probability $P(\mathbf{X}, t + dt|\mathbf{X}_0, t_0)$ of the system being in state $\mathbf{X}(t)$ during time interval dt given $\mathbf{X}(t_0) = \mathbf{X}_0$. This is modelled by a continuous time Markov process. In order to compute this probability, we base on the following observations:

1. For a particular reaction R_j, the probability of the system reaching state $\mathbf{X}(t)$ during time interval dt in which R_j occurs is $a_j(\mathbf{X} - \boldsymbol{\nu}_j)P(\mathbf{X} - \boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0)dt$. By summing over all reactions, $\sum_{j=1}^{M} a_j(\mathbf{X} - \boldsymbol{\nu}_j)P(\mathbf{X} - \boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0)dt$ accounts for the probability of reaching $\mathbf{X}(t)$ by any one reaction in time dt.

2. The probability of the system being in state $\mathbf{X}(t)$ where no reaction occurs during time interval dt is $\left(1 - \sum_{j=1}^{M} a_j(\mathbf{X})dt\right) P(\mathbf{X}, t|\mathbf{X}_0, t_0)$

As a result, we have the following expression for $P(\mathbf{X}, t+dt|\mathbf{X}_0, t_0)$:

$$P(\mathbf{X}, t+dt|\mathbf{X}_0, t_0) = \sum_{j=1}^{M} a_j(\mathbf{X}-\boldsymbol{\nu}_j)P(\mathbf{X}-\boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0)dt + \left(1 - \sum_{j=1}^{M} a_j(\mathbf{X})dt\right) P(\mathbf{X}, t|\mathbf{X}_0, t_0) \tag{1.2}$$

hence,

$$\frac{P(\mathbf{X}, t+dt|\mathbf{X}_0, t_0) - P(\mathbf{X}, t|\mathbf{X}_0, t_0)}{dt} = \sum_{j=1}^{M} [a_j(\mathbf{X}-\boldsymbol{\nu}_j)P(\mathbf{X}-\boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0) - a_j(\mathbf{X})P(\mathbf{X}, t|\mathbf{X}_0, t_0)] \tag{1.3}$$

By taking the limit of the left term as $dt \to 0$, we obtain:

$$\frac{\partial}{\partial t}P(\mathbf{X}, t|\mathbf{X}_0, t_0) = \sum_{j=1}^{M} [a_j(\mathbf{X}-\boldsymbol{\nu}_j)P(\mathbf{X}-\boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0) - a_j(\mathbf{X})P(\mathbf{X}, t|\mathbf{X}_0, t_0)] \tag{1.4}$$

The last equation is known as the Chemical Master Equation (CME) which describes the transition probability $P(\mathbf{X}, t|\mathbf{X}_0, t_0)$. In general, this differential equation is too difficult to solve since the state space is very large; therefore, a method called Gillespie algorithm (Gillespie, 1977, 2007) is used to generate trajectories $\mathbf{X}(t)$ whose probabilities $P(\mathbf{X}, t)$ satisfy the Chemical Master Equation. In order to simulate $\mathbf{X}(t)$, the basic idea of Gillespie's approach is to determine when the next reaction will occur and what reaction is going to take place. The time for the next reaction to occur is sampled from an exponential distribution and based on the reaction constants. In particular, Gillespie's algorithm is performed in two steps:

1. The time τ until the next reaction will occur is randomly chosen from an exponential distribution with mean $1/a_0(\mathbf{X})$, where $a_0(\mathbf{X}) = \sum_{j=1}^{M} a_j(\mathbf{X})$.

2. The next reaction R_μ which has to occur in the next infinitesimal time interval $(t+\tau, t+\tau+d\tau)$ is randomly chosen with probability $a_\mu d\tau$.

Those two steps of the algorithm will accurately keep track the system's trajectories in time. Indeed, this can be explained as follows, the probability that the next reaction will occur in the infinitesimal time interval $(t+\tau, t+\tau+d\tau)$, and will be R_μ is defined as $P(\tau, \mu)d\tau = P^0(\tau)a_\mu d\tau$. Here $P^0(\tau)$ is the probability that no reaction occurs in the time interval $(t, t+\tau)$, and $a_\mu d\tau$ is the probability that the next reaction R_μ which has to occur in the next infinitesimal time interval $(t+\tau, t+\tau+d\tau)$. In order to compute this probability, we divide the interval time $(t, t+\tau)$ into n sub-intervals of width $\epsilon = \frac{\tau}{n}$. Hence, the probability that none of the reactions occurs in any of n sub-intervals is $\left(1 - \sum_{j=1}^{M} a_j(\mathbf{X})\epsilon\right)^n$. We now take the limit of n to infinite to obtain the probability

that no reaction occurs in the time interval $(t, t+\tau)$ as follows:

$$\begin{aligned} P^0(\tau) &= \lim_{n\to\infty} \left(1 - \sum_{j=1}^{M} a_j(\mathbf{X})\epsilon\right)^n \\ &= \lim_{n\to\infty} \left(1 - \frac{\sum_{j=1}^{M} a_j(\mathbf{X})\tau}{n}\right)^n \\ &= \exp\left(-\sum_{j=1}^{M} a_j(\mathbf{X})\tau\right) \end{aligned} \tag{1.5}$$

therefore, we get:

$$\begin{aligned} P(\tau,\mu)d\tau &= P^0(\tau)a_\mu d\tau \\ &= \exp\left(-\sum_{j=1}^{M} a_j(\mathbf{X})\tau\right) a_\mu d\tau \\ &= \left(a_0(\mathbf{X})e^{-a_0(\mathbf{X})\tau}d\tau\right)\left(\frac{a_\mu}{a_0(\mathbf{X})}\right) \qquad (1.6) \\ &= P(\tau)P(\mu|\tau) \qquad (1.7) \end{aligned}$$

here, $P(\tau) = a_0(\mathbf{X})e^{-a_0(\mathbf{X})\tau}d\tau$ is the probability of the waiting time that the next reaction occur; $P(\mu|\tau) = \frac{a_\mu}{a_0(\mathbf{X})}$ is the probability that given τ, reaction μ is chosen to occur. It is clear that τ is an exponential distributed variable with mean $1/a_0(\mathbf{X})$; therefore, τ can be chosen from an exponential distribution with mean $1/a_0(\mathbf{X})$. The next reaction μ is then picked up with probability $\frac{a_\mu}{a_0(\mathbf{X})}$. Consequently, the two random variables τ and μ are chosen as follows:

$$\tau = \frac{1}{a_0(\mathbf{X})}\ln\left(\frac{1}{r_1}\right),\ \mu = \text{the integer satisfying} \sum_{j=1}^{\mu-1} a_j(\mathbf{X}) < r_2 \sum_{j=1}^{M} a_j(\mathbf{X}) \leq \sum_{j=1}^{\mu} a_j(\mathbf{X}) \tag{1.8}$$

here, r_1, r_2 are two random numbers generated from the uniform distribution in the unit interval. After these two steps, the time is set to $t+\tau$ and the state $\mathbf{X}$ is updated to $\mathbf{X} + \boldsymbol{\nu}_\mu$.

As we can see, the CME is used for stochastic molecular details and can be simulated by using Gillespie's algorithm. However, it is still complicated to understand the underlying behaviour of the system. As a consequence, it should be reduced to an Ordinary Differential Equation (ODE) which is simpler and can be used for dynamical systems analysis. Since the ODEs tracks the time evolution of the system where all fluctuations are ignored; therefore, it is simpler to use the ODEs to explicitly determine the state space of the system. In order to do that simplification, we need to study the Langevin approximation which is seen as a bridge between the ODE and CME. A description of the Langevin approximation is detailed in the following section.

1.1.3 The Langevin Approximation

In this section, we will try to approximate the discrete Markov process defined by the CME (1.4) by a continuous Markov process by replacing a Poisson distribution by a Normal distribution. Firstly, let assume that we can find τ small enough such as no propensity function $a_j(\mathbf{X})$ is likely to change its value by a significant amount in the time interval $[t, t+\tau)$ for all $1 \leq j \leq M$. Consequently, the number of times reactions R_j occur in the interval has a Poisson distribution with parameter $a_j(\mathbf{X})\tau$ (Gillespie, 2007). Therefore, the molecular numbers of species can be updated according to the following equation:

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \mathcal{P}_j(a_j(\mathbf{X})\tau)\boldsymbol{\nu}_j, \tag{1.9}$$

Here $\mathcal{P}_j(.)$ is a Poisson distribution with the same mean and variance $a_j(\mathbf{X})\tau$. Supposing that τ is also large enough that the average number of reaction firing during τ is $\gg 1$:

$$a_j(\mathbf{X})\tau \gg 1 \quad \text{for all } j = 1, \ldots, M. \tag{1.10}$$

Under these assumptions, we can approximate the Poisson distribution by a normal (Gaussian) distribution $\mathcal{N}(m, \sigma^2)$ with the same mean and variance. As a result, the equation now describes the time evolution of the system in terms of continuous random variables rather than discrete variables:

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \mathcal{N}_j(a_j(\mathbf{X})\tau, a_j(\mathbf{X})\tau)\boldsymbol{\nu}_j = \mathbf{X}(t) + \sum_{j=1}^{M} \left[a_j(\mathbf{X})\tau + \sqrt{a_j(\mathbf{X})\tau}\mathcal{N}_j(0,1) \right] \boldsymbol{\nu}_j$$

Thus

$$\mathbf{X}(t+\tau) = \mathbf{X}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j a_j(\mathbf{X})\tau + \sum_{j=1}^{M} \boldsymbol{\nu}_j \sqrt{a_j(\mathbf{X})}\, dW_j \tag{1.11}$$

where $dW_j = \mathcal{N}_j(0,1)\sqrt{\tau}$ is a *Wiener process* or *Brownian motion* (Dehling et al., 2007). This equation is known as a Chemical Langevin Equation (CLE) or Langevin approximation. In order to describe the CLE in terms of molecular concentration, the propensities $a_j(\mathbf{X})$ should be appropriately replaced by $\widetilde{a}_j(\mathbf{Y})$ scaled by powers of volume Ω to translate between molecule numbers $\mathbf{X}$ and concentrations $\mathbf{Y} = \mathbf{X}/\Omega$. As a result, in general, we have $\widetilde{a}_j(\mathbf{Y}) = a_j(\mathbf{X})/\Omega$ (Gillespie, 2000; Pakka et al., 2010). Consequently, equation (1.9) can be re-written as:

$$\begin{aligned} \mathbf{Y}(t+\tau) &= \mathbf{Y}(t) + \frac{1}{\Omega}\sum_{j=1}^{M} \mathcal{N}_j(\Omega\widetilde{a}_j(\mathbf{Y})\tau, \Omega\widetilde{a}_j(\mathbf{Y})\tau)\boldsymbol{\nu}_j \\ &= \mathbf{Y}(t) + \frac{1}{\Omega}\sum_{j=1}^{M} \left[\Omega\widetilde{a}_j(\mathbf{Y})\tau + \sqrt{\Omega}\sqrt{\widetilde{a}_j(\mathbf{Y})\tau}\mathcal{N}_j(0,1) \right] \boldsymbol{\nu}_j \end{aligned} \tag{1.12}$$

Therefore,

$$\mathbf{Y}(t+\tau) = \mathbf{Y}(t) + \sum_{j=1}^{M} \nu_j \widetilde{a}_j(\mathbf{Y})\tau + \frac{1}{\sqrt{\Omega}} \sum_{j=1}^{M} \boldsymbol{\nu}_j \sqrt{\widetilde{a}_j(\mathbf{Y})}\, dW_j \tag{1.13}$$

1.1.4 The RRE Approximation

In equation (1.13), as we take the limit of volume $\Omega \to \infty$ and keep the concentration $\mathbf{Y}$ constant, the second term in the right side of the equation will become negligibly small compared with the other terms; therefore, the CLE reduces to the following equation (Gillespie, 2000):

$$\mathbf{Y}(t+\tau) = \mathbf{Y}(t) + \sum_{j=1}^{M} \boldsymbol{\nu}_j \widetilde{a}_j(\mathbf{Y})\tau \tag{1.14}$$

Therefore

$$\frac{d\mathbf{Y}(t)}{dt} = \sum_{j=1}^{M} \boldsymbol{\nu}_j \widetilde{a}_j(\mathbf{Y}) \tag{1.15}$$

The last equation is known as the "concentration" form, a *reaction rate equation* (RRE). We can take the equation (1.1) in the previous section as an example; the corresponding reaction rate equation for concentration variable $\mathbf{Y} = (Y_1, Y_2)^\intercal$ is given as follows:

$$\begin{aligned} &\frac{dY_1}{dt} = k_1 Y_1 Y_2 - k_2 {Y_1}^2 \\ &\text{and} \\ &\frac{dY_2}{dt} = -k_1 Y_1 Y_2 + k_2 {Y_1}^2 \end{aligned} \tag{1.16}$$

where k_1, k_2 are deterministic rate constants and

$$Y_1 = \frac{X_1}{\Omega}, \quad Y_2 = \frac{X_2}{\Omega}, \quad \widetilde{a}_1(\mathbf{Y}) = k_1 Y_1 Y_2, \quad \widetilde{a}_2(\mathbf{Y}) = k_2 {Y_1}^2$$

In order to find the relationship between the deterministic and stochastic rate constants, we can re-write (1.16) in matrix form as follows:

$$\frac{d}{dt}\begin{pmatrix} \mathbf{X} \\ \overline{\Omega} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} k_1 \left(\frac{X_1}{\Omega}\frac{X_2}{\Omega}\right) \\ k_2 \left(\frac{X_1}{\Omega}\right)^2 \end{pmatrix}$$

$$\Rightarrow d(\mathbf{X}) = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \frac{k_1}{\Omega} X_1 X_2 dt \\ \frac{k_2}{\Omega} {X_1}^2 dt \end{pmatrix}$$

On the other hand, we have

$$d(\mathbf{X}) = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} a_1(\mathbf{X})dt \\ a_2(\mathbf{X})dt \end{pmatrix}$$

where

$$\begin{aligned} a_1(\mathbf{X}) &= c_1 X_1 X_2 \\ a_2(\mathbf{X}) &= c_2 \frac{X_1(X_1-1)}{2} \approx \frac{c_2}{2} {X_1}^2 \quad \text{for} \quad X_1 \gg 0 \end{aligned} \tag{1.17}$$

By performing propensity function matching, we obtain

$$c_1 = \frac{k_1}{\Omega}, \quad c_2 = \frac{2k_2}{\Omega} \tag{1.18}$$

In our study, the RRE is used for the deterministic models and we also use both Gillespie and Langevin simulations to describe the stochastic dynamical behaviour of the system. The continuous stochastic model built up by using Langevin approximation is called a Langevin model which is then simulated and compared to the Gillespie simulation.

In fact, experiments can feed into description of reactions that are expressed in terms of distribution or histogram; therefore, this can be obtained from the Fokker-Planck equation in which we can quantitatively capture the probability distribution of the system states. A description of the Fokker-Planck equation is detailed in the following section.

1.1.5 The Fokker-Planck Equation

The Fokker-Planck equation describes the time evolution of the probability density function of species motion in stochastic dynamic systems. In deterministic systems, the system state assumes a well-defined value at a unit time for a given set of initial conditions. In stochastic systems, however, it is a random variable which is characterized by its time-parameterized probability density function (PDF), $P(\mathbf{X}, t)$. I will show how the Fokker-Planck Equation can be obtained from the Chemical Master Equation. Indeed, assuming that $\mathbf{X}(t)$ is a vector of the number of molecules, then the time-evolution description of the joint probability distribution of all species is given by the following Chemical Master Equation (CME):

$$\frac{\partial}{\partial t} P(\mathbf{X}, t) = \sum_{j=1}^{M} \left[P\left(\mathbf{X} - \boldsymbol{\nu}_j, t\right) a_j\left(\mathbf{X} - \boldsymbol{\nu}_j\right) - P\left(\mathbf{X}, t\right) a_j\left(\mathbf{X}\right) \right] \tag{1.19}$$

We now set $f_j(\mathbf{X}, t) = P\left(\mathbf{X}, t\right) a_j\left(\mathbf{X}\right)$ and assume that we will treat $\mathbf{X}$ as real numbers instead of integers, we expect that this assumption is true if the molecule numbers are large enough. We also assume that functions $f_j(\mathbf{X}, t)$ are infinitely differentiable, and

therefore satisfies the following Taylor expansion:

$$f_j(\mathbf{X}-\boldsymbol{\nu}_j,t)-f_j(\mathbf{X},t)=\sum_{i=1}^{N}(-\nu_{ij})\frac{\partial}{\partial X_i}f_j(\mathbf{X},t)+\frac{1}{2}\sum_{i=1}^{N}\frac{\partial^2}{\partial {X_i}^2}\left({\nu_{ij}}^2 f_j(\mathbf{X},t)\right)+\sum_{\substack{i,k=1\\ i\neq k}}^{N}\frac{\partial^2}{\partial X_i\partial X_k}\left(\nu_{ij}\nu_{kj}f_j(\mathbf{X},t)\right)$$

therefore,

$$\begin{aligned}P\left(\mathbf{X}-\boldsymbol{\nu}_j,t\right)a_j\left(\mathbf{X}-\boldsymbol{\nu}_j,t\right)-P\left(\mathbf{X},t\right)a_j\left(\mathbf{X}\right)=&-\sum_{i=1}^{N}\frac{\partial}{\partial X_i}\left[\left(\nu_{ij}a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]\\&+\frac{1}{2}\sum_{i=1}^{N}\frac{\partial^2}{\partial {X_i}^2}\left[\left({\nu_{ij}}^2a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]\\&+\sum_{\substack{i,k=1\\ i\neq k}}^{N}\frac{\partial^2}{\partial X_i\partial X_k}\left[\left(\nu_{ij}\nu_{kj}a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]\end{aligned}\tag{1.20}$$

Summing over all the reactions, this yields:

$$\begin{aligned}\frac{\partial}{\partial t}P(\mathbf{X},t)=&-\sum_{i=1}^{N}\frac{\partial}{\partial X_i}\left[\left(\sum_{j=1}^{M}\nu_{ij}a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]+\frac{1}{2}\sum_{i=1}^{N}\frac{\partial^2}{\partial {X_i}^2}\left[\left(\sum_{j=1}^{M}\nu_{ij}^2a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]\\&+\sum_{\substack{i,k=1\\ i\neq k}}^{N}\frac{\partial^2}{\partial X_i\partial X_k}\left[\left(\sum_{j=1}^{M}\nu_{ij}\nu_{kj}a_j(\mathbf{X})\right)P(\mathbf{X},t)\right]\end{aligned}\tag{1.21}$$

The last equation is a description of the *Chemical Fokker-Planck Equation* (Gillespie, 2002) , I now show how this equation is related to the Chemical Langevin Equation. To do so, we re-write the Langevin equation as follows:

$$\mathbf{X}(t+\tau)=\mathbf{X}(t)+\sum_{j=1}^{M}\boldsymbol{\nu}_j a_j(\mathbf{X})\tau+\sum_{j=1}^{M}\boldsymbol{\nu}_j\sqrt{a_j(\mathbf{X})}\,dW_j\tag{1.22}$$

Where $dW_j=\mathcal{N}_j(0,1)\sqrt{\tau}$ is a Wiener process. This equation can be described as a stochastic process with the drift term $\boldsymbol{\mu}(\mathbf{X},t)$ and the diffusion term $\boldsymbol{\sigma}(\mathbf{X},t)$ as follows:

$$d\mathbf{X}(t)=\boldsymbol{\mu}(\mathbf{X},t)dt+\boldsymbol{\sigma}(\mathbf{X},t)\,dW\tag{1.23}$$

where $\boldsymbol{\mu}(\mathbf{X},t)$ is a vector with elements $\mu_i(\mathbf{X},t)=\sum_{j=1}^{M}\nu_{ij}a_j(\mathbf{X})$, $\boldsymbol{\sigma}(\mathbf{X},t)$ is a matrix with components $\sigma_{ij}(\mathbf{X},t)=\nu_{ij}\sqrt{a_j(\mathbf{X})}$. As a result, the corresponding Fokker-Planck

is expressed as follows:

$$\frac{\partial}{\partial t}P(\mathbf{X},t) = -\sum_{i=1}^{N}\frac{\partial}{\partial X_i}\left(\mu_i(\mathbf{X},t)P(\mathbf{X},t)\right)+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2}{\partial X_i \partial X_j}\left(\sum_{k=1}^{M}\sigma_{ik}(\mathbf{X},t)\sigma_{jk}(\mathbf{X},t)P(\mathbf{X},t)\right) \tag{1.24}$$

Again, we take equation (1.1) for an example, the corresponding Fokker-Planck equation for the system $\mathbf{X} = [X_1, X_2]^\intercal$ is described as follows:

$$\frac{\partial}{\partial t}P(\mathbf{X},t) = -\sum_{i=1}^{2}\frac{\partial}{\partial X_i}\left(\mu_i(\mathbf{X},t)P(\mathbf{X},t)\right)+\frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\partial^2}{\partial X_i \partial X_j}\left(\sum_{k=1}^{2}\sigma_{ik}(\mathbf{X},t)\sigma_{jk}(\mathbf{X},t)P(\mathbf{X},t)\right) \tag{1.25}$$

where

$$a_1(\mathbf{X}) = c_1 X_1 X_2 = \frac{k_1 X_1 X_2}{\Omega}, \quad a_2(\mathbf{X}) = c_2\frac{X_1(X_1-1)}{2} = \frac{k_2 X_1(X_1-1)}{\Omega} \quad \text{(see equation (1.18))}$$

$$\boldsymbol{\mu}(\mathbf{X},t) = \begin{pmatrix}\mu_1(\mathbf{X},t)\\ \mu_2(\mathbf{X},t)\end{pmatrix}, \quad \mu_1(\mathbf{X},t) = a_1(\mathbf{X}) - a_2(\mathbf{X}), \quad \mu_2(\mathbf{X},t) = -a_1(\mathbf{X}) + a_2(\mathbf{X})$$

$$\boldsymbol{\sigma}(\mathbf{X},t) = \begin{pmatrix}\sqrt{a_1(\mathbf{X})} & -\sqrt{a_2(\mathbf{X})}\\ -\sqrt{a_1(\mathbf{X})} & \sqrt{a_2(\mathbf{X})}\end{pmatrix}$$

1.1.6 Model Reduction

The Gillespie's algorithm provides a means to simulate the stochastic dynamical behaviour of species in biochemical systems. This allows us to obtain a sample of the trajectories, however, it does not help understand the underlying behaviour which is encapsulated in the set of reaction equations. The CLE is an approximation of the Gillespie simulation where the jump Markov process is replaced by a continuous Markov process. The Fokker-Planck equation which is the direct consequence of the CLE, on the other hand, probabilistically captures the motion of the system. Unlike these methods, the RRE only observes the mean change in the number of components which is therefore simpler to analyse. However, as the number of variables is large, the RRE description would be hard to understand the complex interaction among those variables. In contrast, if the behaviour of the system is somehow described by a lower dimensional model, says a two-dimensional model, we then can mathematically analyze the dynamics of the system by plotting the time evolution of the two variables on a plane. The impact of the two variables on the dynamical behaviour is therefore easier to analyse. As a result, in order to better understand the underlying behaviour of the system, we need to do a model reduction so that the very complicated system can be reduced to a simpler system which is potentially easier. In fact, this can be done by replacing the CME by the

reaction rate equations. Next, the differential equations can be reduced by identifying fast processes which are observed to happen very quickly compared to the others in the system. A standard method of doing this is to apply an adiabatic approximation for the fast processes. The description of this method is illustrated in the following section.

1.1.7 Adiabatic Approximation

Firstly, the fundamental principle of the reduction method is to divide the system into two parts which evolve on slow and fast times-scales separately. For example, assuming that we have a fast-slow system of two species as follows :

$$\begin{aligned} \frac{dx}{dt} &= f(x, y, \epsilon) \\ \epsilon\frac{dy}{dt} &= g(x, y, \epsilon) \end{aligned} \tag{1.26}$$

Where $0 < \epsilon \ll 1$ is the singular perturbation parameter (Holmes, 2009). In this example, x is a slow variable whereas y is a fast variable since the rate of change in y is faster than that in x. Such systems are said to be singularly perturbed (Tian, 2003) in which the trajectories follow the slow manifolds that are invariant under the dynamics of the systems. In our particular system, the slow manifold can be obtained by taking the limit of ϵ to zero, then we get:

$$\lim_{\epsilon\to 0} \frac{\epsilon dy}{dt} = \lim_{\epsilon\to 0} g(x, y, \epsilon) = 0 \tag{1.27}$$

Assuming that there exists $y = h_0(x)$ such that $g(x, h_0(x), 0) = 0$, we say our system is now reduced to a slow manifold. In this case, the second reaction is actually a fast process which is then eliminated off the system. This is known as an *adiabatic approximation.* As a result, the adiabatic approximation can be obtained by simply setting $\epsilon = 0$. In other words, the fast processes are set to their steady states so that the rest the system only responds to the evolution of the slow processes.

In fact, there is not only one slow manifold $h_0(x)$ which exists, but a family of slow manifolds $h(x, \epsilon)$ that drive the system (Zagaris et al., 2004; Sobolev and Tropkina, 2012). To find those slow manifolds, we utilize an approach called *invariant manifold method* which is suggested by Roussel and Fraser (Roussel, 1997; Roussel and Fraser, 2001). This method is described in the following section.

1.1.8 Invariant Manifold Method

The basic idea of this method is that, in order to find the slow manifolds, we won't set ϵ to zero but do the asymptotic expansions for $h(x, \epsilon)$ as follows:

$$h(x, \epsilon) = h_0(x) + \epsilon h_1(x) + \epsilon^2 h_2(x) + \dots \tag{1.28}$$

Therefore, from (1.26) we obtain:

$$g(x, h(x, \epsilon), \epsilon) = \epsilon \frac{dy}{dt} = \epsilon \frac{dy}{dx}.\frac{dx}{dt} = \epsilon \frac{\partial h(x, \epsilon)}{\partial x} f(x, h(x, \epsilon), \epsilon) \tag{1.29}$$

hence,

$$g(x, h(x, \epsilon), \epsilon) - \epsilon \frac{\partial h(x, \epsilon)}{\partial x} f(x, h(x, \epsilon), \epsilon) = 0 \tag{1.30}$$

This equation can be solved to obtain the following general expression of $h(x, \epsilon)$:

$$h(x, \epsilon) = F(x, \partial h(x, \epsilon)/\partial x, \epsilon) \tag{1.31}$$

In order to solve equation (1.31), we start with an initial function $h^0(x, \epsilon = 0)$ and putting it back into the right term of the equation to obtain $h^1(x, \epsilon)$. This procedure is then repeated until convergence. The detailed description of this method will be presented in Chapter 3.

1.2 Motivation

Modelling GRNs is challenging on many different levels. Empirically, it is difficult to measure which reactions occur in a living organism. Models are therefore, by necessity, built up by guess work based on the basic mechanism for protein production and references made from a series of indirect experiments to identify proteins which appear to play a key role in a mechanism of interest. The models derived are usually a set of reactions together with best guessed (or tuned) reaction rates that form the *Chemical Master Equations*.

The behaviour of the system can then be simulated using the Gillespie algorithm which allows those who construct these models to investigate questions such as the robustness to changes in parameters. However, understanding the dynamics directly from the Chemical Master Equations is usually difficult due to the large number of chemical species and the lack of analytic tools that can be directly applied to a set of chemical reactions. In order to overcome this, the common practice is to try to replace the chemical master equations by a more analytically amenable model which is normally described by differential equations or stochastic differential equations. The next step is to eliminate fast reacting components in order to simplify the model. In the chapters on the wild-

type system and in simplification (Chapters 2, 3, 4, 5), I start with the CME which I take as the generator of the correct behaviour, and then use the RRE, Langevin and the Fokker-Planck equations of reduced models to approximate this reference behaviour. In Chapter 6, I reverse the order of introducing the models, I start with a RRE and infer the necessary CME in order to generate a reference stochastic behaviour of a synthetic circuit.

I will concentrate on modelling the competence behaviour displayed by *B. Subtilis* and in particular, the model proposed by Süel et al. (2006, 2007). This is a model that has received very considerable attention (792 references to date). Their paper provided a first step towards simplifying the chemical master equations by replacing the reaction rate equation by differential equations and then eliminating "fast" changing variables to obtain a two species model consisting a pair of coupled differential equations. The authors then studied the structure of the fixed points for these differential equations and showed that a slight modification can lead to a change in the fixed points stability and the system behaviour.

My interest has been to extend the analysis in their papers. In particular, the competence behaviour is stochastically driven, so we would like to put back this stochastic behaviour in the simplified model. As we will see through this thesis this proved far more challenging and a low-dimensional model was not fully able to be constructed using a standard model reduction procedure. This procedure is basically described in the following steps:
1. Replace the chemical master equations by a set of Langevin equations
2. Eliminate fast reactions from the RRE to obtain a reduced set of equations describing the dynamics of slowly varying species.

However, both these steps were problematic. The Langevin equations are stochastic differential equations which capture the mean change in the number of components in a reaction (the drift term) and the fluctuations around this (the noise term). As the chemical master equations can be described by a Poisson process, it is usually expected the drift term to be proportional to the number of reactants and the variance describing the fluctuations to be equal to the drift term. However, for this reduction to be accurate requires the number of the reactants to be sufficient large. Unfortunately, the nature of the competence mechanism is that it relies on the fluctuation of very small number of mRNA molecules described in Süel et al. (2006, 2007), which is then amplified through a positive feedback mechanism. This creates a number of problems for standard modelling. Firstly, the size of fluctuations in the Langevin model can drive the molecule number negative (which is clearly unphysical). Secondly, despite the fact that the mRNA molecules react very quickly, I have failed to find a mathematically consistent way of eliminating the mRNA molecules, while capturing the large fluctuations they induce (these problems will be discussed in Chapter 4). I, thus, have to introduce the fluctuations artificially to model those observed in the full system. Although I was

unable to solve this problem, one of the contribution of the thesis is to pin-point this problem. Moreover, as I show late in Chapter 5, I model the noise by tuning the noise term in a Langevin equation to fit the distribution generated from the CME.

The second difficulty is in eliminating that species involved in fast reactions. In fact, if you are only interested in the fixed-point structure, then you can ignore the noise term and set all the drift term to zero. However, we can do better than that. The competence behaviour is captured by a slow manifold. This manifold can be projected onto the ComK-ComS plane where it is seen that the trajectories of Gillespie simulation follow roughly the same trajectory. The reason for this is because the dynamics of the whole system breaks up into different time scales. Thus, although the individual reactions do not have very different time scales, there still exists one slow time scale describing the trajectory of the system after the transition to competence. This trajectory is uniquely characterized by the values of ComK and ComS. However, as I will show, to obtain the correct trajectory, it is insufficient to do the reduction as described by Süel et al. (2007), but instead use an *invariant manifold method* first proposed by Roussel (1997); Roussel and Fraser (2001). A second contribution of the thesis is to show how to apply this to the competence model (this is described in section 3.2 of Chapter 3).

Once we have an accurate 2-dimensional model of the dynamics, it is interesting to put back the stochastic behaviour to obtain a full low-dimensional description of the behaviour of the system. Unfortunately, as I discussed earlier, I have been unable to systematically derive a correct stochastic model. However, we can build a model with a tunable noise chosen to reproduce the noise observed in the full system. This can be done by exploring the empirical noise during the simulation and approximating the relation between the size of noise and the corresponding concentration of reactant. In spite of the fact that I could not provide an explicit mathematical description of this relation, however, I would show evidence that our effective model of the fluctuations in a Langevin model reproduces the dynamical behaviour of the original system. This is another critical contribution to the thesis. When the stochastic reduced model is obtained, then we can have a number of available analytic tools allow us to investigate the behaviour better. Particularly, we can use the Fokker-Planck equation to obtain a probability density function which describes the probability of the system being at a particular state. I describe this in Chapter 5.

I have also extended this analysis to a second set of models produced by the same group and described in Cagatay et al. (2009a). Here the authors used techniques from synthetic biology to construct a strain of cells containing different circuits for becoming competent. The group's starting point was a set of differential equations with the same qualitative behaviour as the wild-type. I analysed this new model and show that their differential equations are inconsistent with a physically meaningful set of Chemical Master Equations. I also show that their model is susceptible to being trapped around the secondary fixed point (which is weakly stable in their model), giving rise to transient

noise oscillatory behaviour. Interestingly, the second fixed point can be avoided by reducing the noise, stopping the system from jumping into the basin of attraction of this fixed point. I provide this analysis in Chapter 6.

To sum up, the regulatory network we have been working on is more complicated than it has been thought to be. The more we analyze the system, the more interesting results we can find. This is also the reason why we have spent lots of time trying to understand the underlying behaviour of the system. Thus, the discovery of negative results has prevented us from following a straightforward publishing path. However, since we have found useful evidences which help change the way we understand the wild-type genetic circuit, we have submitted a paper for publishing those results.

1.3 Structure of Thesis

The rest of thesis is structured as follows. In Chapter 2, I describe the competence circuit which is used to model cellular behaviours followed by reproducing the phase-plane analysis of this circuit using the discrete and continuous models. I also describe the model in more detailed way by giving a set of differential equations which are derived from the Chemical Master Equation (CME) for biochemical reactions in gene expression. In Chapter 3, I show the inadequacy of the assumptions made by Süel et al. (2007) which leads to poor quantitative observations of the system. To solve this problem, I present a solution to the problem of dimensionality reduction based on invariant slow manifold technique. This allows our original high-dimensional system to be reduced to a lower-dimensional system which is easier for a systematic analysis. I also make a comparison between the novel reduced model with the original two-dimensional model with respect to the full-dimensional system. In the competence regime, the reduced model accurately captures the time scale for returning to the vegetative state, and is thus an improvement upon the analysis of Süel et al. (2007).

In fact, the dynamics of the system is driven by noise which can be described by a stochastic process. As a result, I demonstrated standard methods of approximating the model and putting the noise back into the reduced system to be able to come up with a reduced stochastic model which provides a correct way of simulating the dynamics of system.

However, as I will show that this step is a bit tricky and it is very difficult to capture the noise of the system. In order to really understand the system, Chapter 4 is followed by conducting a small research about a simple two species genetic circuit which is narrowed down from our complicated high dimensional system. The purpose of this study is to focus on a still unresolved problem in model reduction which currently prevent us from accurately reducing the full stochastic system to a low dimensional stochastic system.

By focusing on a simpler problem (only 2 variables) we seek to reveal the principles for such model reduction.

Finally, by adding some noise to the continuous model, the stochastic continuous model can be used to stimulate the cell behaviour more realistically. As a result, a Fokker-Planck equation that accurately captures small fluctuations about the stable attractor of the system will be detailed in Chapter 5. We also apply this equation in order to mathematically approximate the probability distribution of molecular concentrations as cells make a transition from their original vegetative state into a competent state followed by an estimation of the initialization probability of entrance into competence. In Chapter 6, we study another genetic circuit called SynExSlow which was introduced by Cagatay et al. (2009a). This circuit was expected to produce a similar behaviour compared to the wild-type but provide less variability in competence duration. In order to understand the underlying behaviour, the authors tried to reduce the system to a low-dimensional system from the chemical reactions. However, as we show that the RRE of the model can not be reconstructed from a physically allowed Gillespie simulation; therefore, it is needed to be simulated by a stochastic Langevin approximation. We re-engineered a valid CME so that the mean dynamics could possibly be tracked by the same RRE as in Cagatay et al. (2009a). Nevertheless, we will show that the stochastic Langevin approximation of the SynExSlow model does not produce the same behaviour in comparison with Gillespie simulation.

A large part of these sections has been retrieved from Süel et al. (2007) involving hypothesis and parameters of the models which will be used through the thesis and future work. Eventually, the conclusions and future work will complete the report in Chapter 7.

Chapter 2

Competence Circuit

In this chapter, we will study a particular genetic circuit for understanding the competence phenomenon in *B. subtilis*. The main purpose of this chapter is to reproduce the results of Süel et al. (2007). However, we note a discrepancy in their results in Figure 2.10 which will tie into our further analysis in Chapter 3. As mentioned earlier, the development of genetic competence is controlled by the competence-specific transcription factor ComK. In our circuit, ComK activates transcription of its own gene while being controlled by other genes including *mecA*, *clpC*, and *comS*. ComK forms a ternary complex with ClpC and MecA that prevents it from binding to its specific DNA target, thus preventing its own activation, ComS causes the reactivation of ComK by dissociating the ternary complex allowing ComK to bind to its specific DNA target. In the following section, we will briefly describe the competence circuit introduced by Süel et al. (2006, 2007).

2.1 Competence Circuit Architecture

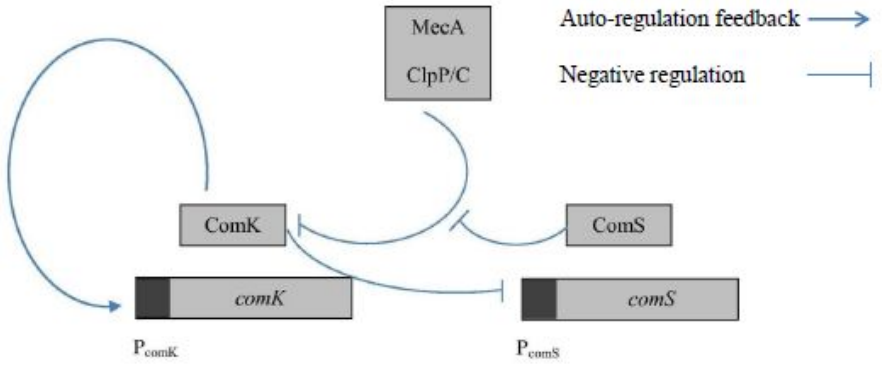


FIGURE 2.1: Competence circuit architecture.

The circuit architecture basically includes interactions among genetic elements which are depicted in Figure 2.1. As we can see in the figure, the competence circuit includes the following components: two genes *comK* and *comS* corresponding to two proteins ComK and ComS, respectively; and promoters P_{comK} and P_{comS}. In the figure, ComK activates the expression of its own gene (auto-regulation feedback) and inhibits expression of ComS (negative regulation), that in turn interferes with degradation of ComK. The complex of MecA, ClpP/C also actively degrades ComK. In the following section, we will describe how to represent the relationship of these molecular components in an explicit form.

2.2 Modelling of Competence Circuit

2.2.1 Discrete Stochastic Model

The natural competence happens under environmental conditions as discussed previously. However, it does not mean that all cells become competent under the same conditions. In fact, there are some cells that make a transition to competent state while others may not. In other words, the competence should be described as random events in which there is a switching state where the transition can be made by some stochastic factor. As a result, if the switch is stochastic then different cells become competent at different times which might have a selective advantage for the population. From this point of view, a discrete stochastic model has been built in order to represent and simulate the behaviour of the competence circuit by means of stochastic biochemical reactions.

The expression of ComK is described by the following reactions (we denote square bracket for molar concentration of reactant):

$$\begin{aligned} \mathrm{P}^{\mathrm{const}}_{\mathrm{comK}} &\xrightarrow{k_1} \mathrm{P}^{\mathrm{const}}_{\mathrm{comK}} + \mathrm{mRNA}_{\mathrm{comK}} \\ \mathrm{P}_{\mathrm{comK}} &\xrightarrow{f([ComK],k_2,k_k,n)} \mathrm{P}_{\mathrm{comK}} + \mathrm{mRNA}_{\mathrm{comK}} \\ \mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_3} \mathrm{mRNA}_{\mathrm{comK}} + \mathrm{ComK} \end{aligned} \tag{2.1}$$

P^{const}_{comK} and P_{comK} are constitutive and regulated promoters of ComK, respectively. $mRNA_{comK}$ and $mRNA_{comS}$ are $mRNA$ molecules from which proteins ComK and ComS are translated, respectively. The symbols above the arrows denote the probabilities of reactions in unit time. The first two reactions represent how much $mRNA_{comK}$ is produced from the binding of protein to the promoters on DNA, this is known as transcriptional regulation. The last reaction shows how much protein ComK is synthesized

from $mRNA_{comK}$, this is known as a translational regulation. The underlying biochemical mechanism (transcription and translation) as well as the regulation of transcription and translation can be found in the Appendix A.1, A.2.

Likewise, the expression of ComS is described as below:

$$\begin{aligned} \mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} &\xrightarrow{k_4} \mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} + \mathrm{mRNA}_{\mathrm{comS}} \\ \mathrm{P}_{\mathrm{comS}} &\xrightarrow{g([ComK],k_5,k_s,p)} \mathrm{P}_{\mathrm{comS}} + \mathrm{mRNA}_{\mathrm{comS}} \\ \mathrm{mRNA}_{\mathrm{comS}} &\xrightarrow{k_6} \mathrm{mRNA}_{\mathrm{comS}} + \mathrm{ComS} \end{aligned} \tag{2.2}$$

In these equations, the regulated transcription rates of ComK and ComS are given by the following Hill equations (see Appendix A.3):

$$f\left([ComK], k_2, k_k, n\right) = \frac{k_2[ComK]^n}{{k_k}^n + [ComK]^n} \quad , \quad g\left([ComK], k_5, k_s, p\right) = \frac{k_5}{1 + \left(\frac{[ComK]}{k_s}\right)^p} \tag{2.3}$$

The first equation represents the auto-regulation feedback of ComK, whereas the second one shows the inhibition (negative regulation) of ComK on ComS.

The Hill equation involves concentrations of ComK and ComS whereas the discrete model is described in terms of number of molecule. Hence, the relationship between concentration and molecular number is captured in the parameter Ω:

$$\begin{aligned} \Omega = AV = 6.023 \times 10^{23} molec/mol \times 1.66 \mu m^3 &= 6.023 \times 10^{23} molec/mol \times 1.66 \times 10^{-15} l \\ &= 0.99982 \times 10^9 molec/M \\ &\approx 1 molec/\, nM \end{aligned} \tag{2.4}$$

Where A is Avogadro's number and V is cell volume which is taken to be $1.66 \mu m^3$. This value is the same as that suggested in Süel et al. (2007). In fact, by measuring molar concentrations in nM (nano-molar) we can ignore the factor Ω since it is very close to 1; therefore, we can treat the concentrations of species in the same way as their molecular number.

In the circuit, mRNA and proteins are also assumed to be degraded linearly:

$$\begin{aligned} \mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_7} \emptyset & \qquad \mathrm{ComK} &\xrightarrow{k_8} \emptyset \\ \mathrm{mRNA}_{\mathrm{comS}} &\xrightarrow{k_9} \emptyset & \qquad \mathrm{ComS} &\xrightarrow{k_{10}} \emptyset \end{aligned} \tag{2.5}$$

k_1	$0.00021875s^{-1}$	k_7	$0.005s^{-1}$	k_{12}	$0.05s^{-1}$	n	2
k_2	$0.1875s^{-1}$	k_8	$10^{-4}s^{-1}$	k_{13}	$4.5 \times 10^{-6}\, nM^{-1}s^{-1}$	p	5
k_3	$0.2s^{-1}$	k_9	$0.005s^{-1}$	k_{-13}	$5 \times 10^{-5}s^{-1}$		
k_4	$0s^{-1}$	k_{10}	$10^{-4}s^{-1}$	k_{14}	$4 \times 10^{-5}s^{-1}$		
k_5	$0.0015s^{-1}$	k_{11}	$2.02 \times 10^{-6}\, nM^{-1}s^{-1}$	k_k	$5000\, nM$		
k_6	$0.2s^{-1}$	k_{-11}	$5 \times 10^{-4}s^{-1}$	k_s	$833\, nM$		

TABLE 2.1: Parameters of the discrete model.

$[P_{comK}^{const}]$	$1\, nM$
$[P_{comS}^{const}]$	$1\, nM$
$[P_{comK}]$	$1\, nM$
$[P_{comS}]$	$1\, nM$
$[mRNA_{comK}]$	$0\, nM$
$[mRNA_{comS}]$	$0\, nM$
$[MecA]$	$23\, nM$
$[MecA_K]$	$0\, nM$
$[MecA_S]$	$477\, nM$
$[ComK]$	$69\, nM$
$[ComS]$	$409\, nM$

TABLE 2.2: Initial conditions.

Finally, the two proteins bind to MecA competitively and get degraded by the protease:

$$\begin{aligned} \mathrm{MecA{+}ComK} &\underset{k_{-11}}{\overset{k_{11}/\Omega}{\rightleftharpoons}} \mathrm{MecA_K} \qquad \mathrm{MecA_K} \xrightarrow{k_{12}} \mathrm{MecA} \\ \mathrm{MecA{+}ComS} &\underset{k_{-13}}{\overset{k_{13}/\Omega}{\rightleftharpoons}} \mathrm{MecA_S} \qquad \mathrm{MecA_S} \xrightarrow{k_{14}} \mathrm{MecA} \end{aligned} \tag{2.6}$$

The reactions on the left represent the interference of MecA to the proteins by binding/unbinding to them, whereas the reactions on the right show the release of MecA from the proteins.

An example of a set of model parameters is given in Table 2.1. The trajectories generated by the discrete model can be simulated by using Gillespie's algorithm (Gillespie, 2007) (I used Dizzy (Ramsey et al., 2005) to generate the trajectories, the Dizzy file description is detailed in Appendix A.11). Figure 2.2 shows such trajectories plotted with initial conditions given in Table 2.2 on a log-scale phase plane (the labels K, S which are denoted for $[ComK]$ and $[ComS]$, respectively, are treated as molecular numbers so that we can plot on a log-scale plane). In this figure, the initial transient has been thrown away in order to capture stable trajectories after a long running time. Even though the simulation shows us the trajectories which describe the behaviour of the system, we need to describe the system in a mathematical way in order to understand the underlying cellular behaviour of the system. In the following section, we will represent the system

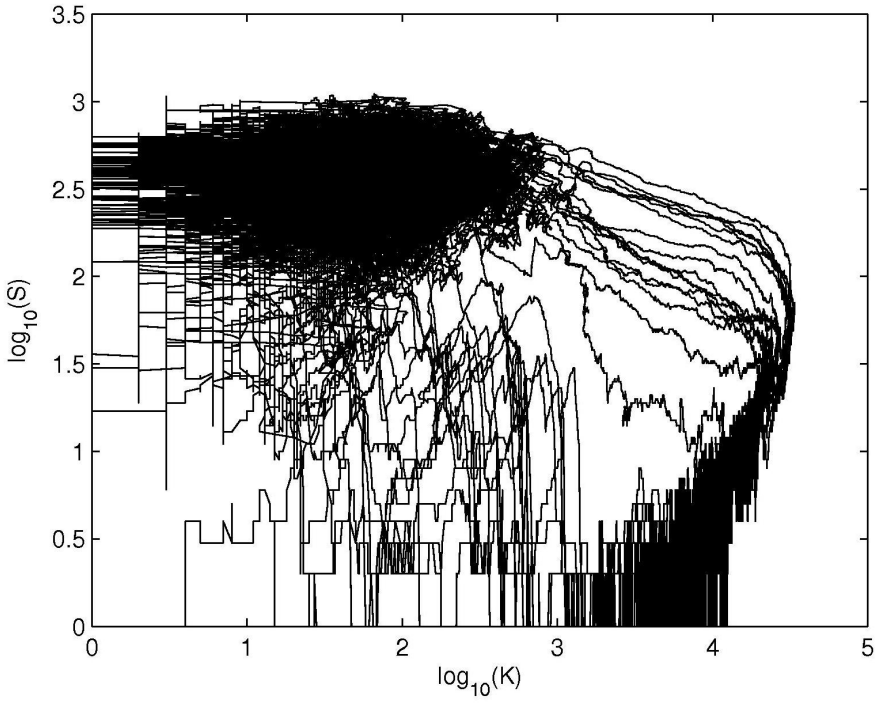


FIGURE 2.2: Trajectories created by the discrete model.

by a set of differential equations.

2.2.2 The Differential Equations

We can approximate the reaction equations (2.1), (2.2), (2.3), (2.5) and (2.6) by a set of differential equations under the assumption of large numbers of molecular populations where we can ignore fluctuations. For example, the rate of change in concentration of mRNA in the reaction (2.5) can be derived as:

$$\frac{d[mRNA_{comK}]}{dt} = -k_7[mRNA_{comK}]$$

As $mRNA_{comK}$ is also involved in reaction (2.1), the final formula for $mRNA_{comK}$ is then:

$$\begin{aligned}\frac{d[mRNA_{comK}]}{dt} &= k_1[P_{comK}^{const}] + f\left([ComK], k_2, k_k, n\right)[P_{comK}] - k_7[mRNA_{comK}] \\ &= k_1[P_{comK}^{const}] + \frac{k_2[ComK]^n}{{k_k}^n + [ComK]^n}[P_{comK}] - k_7[mRNA_{comK}] \qquad (2.7)\end{aligned}$$

Similarly, we have the following differential equations:

$$\frac{d[ComK]}{dt} = -k_{11}[ComK][MecA] + k_{-11}[MecA_K] + k_3[mRNA_{comK}] - k_8[ComK]$$

(2.8)

$$\frac{d[ComS]}{dt} = -k_{13}[ComS][MecA] + k_{-13}[MecA_S] + k_6[mRNA_{comS}] - k_{10}[ComS] \tag{2.9}$$

$$\frac{d[MecA_K]}{dt} = -k_{12}[MecA_K] + k_{11}[ComK][MecA] - k_{-11}[MecA_K] \tag{2.10}$$

$$\frac{d[MecA_S]}{dt} = -k_{14}[MecA_S] + k_{13}[ComS][MecA] - k_{-13}[MecA_S] \tag{2.11}$$

$$\begin{aligned}\frac{d[MecA]}{dt} &= -k_{11}[ComK][MecA] + k_{-11}[MecA_K] + k_{12}[MecA_K] \\ &\quad + k_{14}[MecA_S] - k_{13}[ComS][MecA] + k_{-13}[MecA_S]\end{aligned} \tag{2.12}$$

$$\begin{aligned}\frac{d[mRNA_{comS}]}{dt} &= k_4[P_{comS}^{const}] + g\left([ComK], k_5, k_s, p\right)[P_{comS}] - k_9[mRNA_{comS}] \\ &= k_4[P_{comS}^{const}] + \frac{k_5}{1 + \left(\frac{[ComK]}{k_s}\right)^p}[P_{comS}] - k_9[mRNA_{comS}]\end{aligned} \tag{2.13}$$

This high dimensional system is hard to analyse. In fact, we can eliminate some variables by assuming that very fast processes reaches equilibrium. This is known as a *adiabatic approximation* which has been mentioned in section 1.1.7 of the previous chapter. Although this assumption seems reasonable, we will see that the actual situation is far more subtle. The decay rate of mRNA is higher than that of proteins, and transients in mRNA thus decay much faster than these of proteins. This is the motivation for treating mRNA as a fast variable. As a result, we approximate the effect of mRNA on the protein dynamics by replacing the time-dependent mRNA variable by their steady-state values:

$$\frac{d[mRNA_{comK}]}{dt} = 0 \quad , \quad \frac{d[mRNA_{comS}]}{dt} = 0 \tag{2.14}$$

From (6.17), (2.13) and (2.14) we have:

$$[mRNA_{comK}] = \frac{k_1[P_{comK}^{const}] + \frac{k_2[ComK]^n}{{k_k}^n + [ComK]^n}[P_{comK}]}{k_7} \tag{2.15}$$

$$[mRNA_{comS}] = \frac{k_4[P_{comS}^{const}] + \frac{k_5}{1+\left(\frac{[ComK]}{k_s}\right)^p}[P_{comS}]}{k_9} \tag{2.16}$$

therefore:

$$\begin{aligned}\frac{d[ComK]}{dt} &= -k_{11}[ComK][MecA] + k_{-11}[MecA_K] + k_3[mRNA_{comK}] - k_8[ComK] \\ &= -k_{11}[ComK][MecA] + k_{-11}[MecA_K] + \frac{k_1k_3[P_{comK}^{const}] + \frac{k_2k_3[ComK]^n}{{k_k}^n + [ComK]^n}[P_{comK}]}{k_7} - k_8[ComK]\end{aligned} \tag{2.17}$$

Similarly, we find:

$$\frac{d[ComS]}{dt} = -k_{13}[ComS][MecA] + k_{-13}[MecA_S] + k_6[mRNA_{comS}] - k_{10}[ComS]$$

$$= -k_{13}[ComS][MecA] + k_{-13}[MecA_S] + \frac{k_4 k_6 [P_{comS}^{const}] + \frac{k_5 k_6}{1+\left(\frac{[ComK]}{k_s}\right)^p}[P_{comS}]}{k_9} - k_{10}[ComS] \tag{2.18}$$

The system has now been simplified as a five-dimensional system which includes a tuple of variables $(ComK, ComS, MecA, MecA_K, MecA_S)$. The authors in Süel et al. (2007), however, make it even simpler by assuming that the binding and unbinding processes of proteins and protease complex are very fast so that the rest of the system only responds to the steady-state values of $MecA_K$ and $MecA_S$. Consequently, we set:

$$\frac{d[MecA_K]}{dt} \approx 0 \quad , \quad \frac{d[MecA_S]}{dt} \approx 0 \tag{2.19}$$

From (2.10), (2.11) and (2.19) we obtain:

$$[MecA_K] = \frac{k_{11}[ComK][MecA]}{k_{12} + k_{-11}} \quad , \quad [MecA_S] = \frac{k_{13}[ComS][MecA]}{k_{14} + k_{-13}} \tag{2.20}$$

In addition, the sum $[MecA] + [MecA_K] + [MecA_S]$ remains constant since all reactions involving MecA or its complexes (see Equation (2.6)) conserves this sum. We can see this algebraically by looking at the changes in MecA and its complexes. That means $[MecA] + [MecA_K] + [MecA_S] = M_T = const.$ Plugging this into (6.7), this yields:

$$[MecA] + [MecA_K] + [MecA_S] = \left(\frac{k_{11}[ComK]}{k_{12} + k_{-11}} + \frac{k_{13}[ComS]}{k_{14} + k_{-13}} + 1\right)[MecA] = M_T$$

$$\Rightarrow [MecA] = \frac{M_T}{1 + \frac{k_{11}[ComK]}{k_{12}+k_{-11}} + \frac{k_{13}[ComS]}{k_{14}+k_{-13}}} \tag{2.21}$$

From (2.13), (2.17) and (6.7) we yield:

$$\begin{aligned}
\frac{d[ComK]}{dt} &= -k_{11}[ComK][MecA] + k_{-11}[MecA_K] + \frac{k_1 k_3 [P_{comK}^{const}] + \frac{k_2 k_3 [ComK]^n}{{k_k}^n + [ComK]^n}[P_{comK}]}{k_7} \\
&\quad - k_8[ComK] \\
&= -k_{11}[ComK][MecA] + \frac{k_{11}k_{-11}[ComK][MecA]}{k_{12} + k_{-11}} + \frac{k_1 k_3 [P_{comK}^{const}] + \frac{k_2 k_3}{{k_k}^n + [ComK]^n}[P_{comK}]}{k_7} \\
&\quad - k_8[ComK] \\
&= -\frac{k_{12}k_{11}M_T[ComK]}{(k_{12} + k_{-11})\left(1 + \frac{k_{11}[ComK]}{k_{12}+k_{-11}} + \frac{k_{13}[ComS]}{k_{14}+k_{-13}}\right)} + \frac{k_1 k_3 [P_{comK}^{const}] + \frac{k_2 k_3}{{k_k}^n + [ComK]^n}[P_{comK}]}{k_7} \\
&\quad - k_8[ComK]
\end{aligned} \tag{2.22}$$

Similarly:

$$\frac{d[ComS]}{dt} = -\frac{k_{13}k_{14}M_T[ComS]}{(k_{14}+k_{-13})\left(1+\frac{k_{11}[ComK]}{k_{12}+k_{-11}}+\frac{k_{13}[ComS]}{k_{14}+k_{-13}}\right)} + \frac{k_4k_6[P_{comS}^{const}] + \frac{k_5k_6}{1+\left(\frac{[ComK]}{k_s}\right)^p}[P_{comS}]}{k_9} - k_{10}[ComS] \tag{2.23}$$

As a result, by eliminating some variables which do not significantly impact on the dynamical behaviour of the cells in the long run, we now can describe the system in form of the two differential equations shown as (2.22) and (2.23). This is the two-dimensional continuous model we are going to discuss briefly in the following section.

2.2.3 Analysis of Continuous Model

The dynamics of the circuit is reduced to two differential equations that determine the time evolution of the concentrations ComK and ComS. After redefining some rate equations, Equations (2.22) and (2.23) can be written as:

$$\begin{aligned}\frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K}{1+\frac{K}{\Gamma_k}+\frac{S}{\Gamma_s}} - \lambda_k K \\ \frac{dS}{dt} &= \alpha_s + \frac{\beta_s}{1+\left(\frac{K}{k_s}\right)^p} - \frac{\delta_s S}{1+\frac{K}{\Gamma_k}+\frac{S}{\Gamma_s}} - \lambda_s S\end{aligned} \tag{2.24}$$

where $K = [ComK]$, $S = [ComS]$. In the first equation, α_k and β_k are basal expression rates representing constitutive expression from the promoter P_{comK} and auto-regulated expression rate, respectively. Accordingly, the second term represents the positive transcriptional auto-regulation by ComK of its own gene. In both equations, the third and fourth terms represent the competitive binding of ComK and ComS to MecA and linear degradation with coefficients λ_k and λ_s, respectively.

Here, k_k and k_s are (activation and repression) coefficients which identify concentration of ComK for which its own activation (repression) is half-maximal. In order to make the equation look simpler, we rescale the variables as follows:

$$K \mapsto \frac{K}{\Gamma_k}, \quad S \mapsto \frac{S}{\Gamma_s}, \quad t \mapsto \delta_k t$$

where $\delta_k = \delta_s = \delta$. Dropping primes for simplicity and redefining K, S, t, we come up with the following differential equations:

$$\begin{aligned}\frac{dK}{dt} &= a_k + \frac{b_k K^n}{{k_0}^n + K^n} - \frac{K}{1+K+S} - \Delta_k K \\ \frac{dS}{dt} &= a_s + \frac{b_s}{1+\left(\frac{K}{k_1}\right)^p} - \frac{S}{1+K+S} - \Delta_s S\end{aligned} \tag{2.25}$$

where

$$\begin{aligned} a_k &= \frac{\alpha_k}{\Gamma_k \delta_k}, \quad b_k = \frac{\beta_k}{\Gamma_k \delta_k}, \quad k_0 = \frac{k_k}{\Gamma_k}, \quad \Delta_k = \frac{\lambda_k}{\delta_k} \\ a_s &= \frac{\alpha_s}{\Gamma_s \delta_s}, \quad b_s = \frac{\beta_s}{\Gamma_s \delta_s}, \quad k_1 = \frac{k_s}{\Gamma_k}, \quad \Delta_s = \frac{\lambda_s}{\delta_s} \end{aligned} \tag{2.26}$$

In order to relate the parameters of the continuous model with the reaction rates used in the discrete model, we replace $[ComK]$, $[ComS]$ in (2.22) and (2.23) by K and S, respectively. By comparing the terms in the discrete model with the corresponding terms in the continous model, we obtain:

$$\begin{aligned} \alpha_k &= \frac{k_1 k_3}{k_7}[P_{comK}^{const}] = \frac{k_1 k_3}{k_7}\frac{P_{comK}^{const}}{\Omega} & \beta_k &= \frac{k_2 k_3}{k_7}[P_{comK}] = \frac{k_2 k_3}{k_7}\frac{P_{comK}}{\Omega} \\ \alpha_s &= \frac{k_4 k_6}{k_9}[P_{comS}^{const}] = \frac{k_4 k_6}{k_9}\frac{P_{comS}^{const}}{\Omega} & \beta_s &= \frac{k_5 k_6}{k_9}[P_{comK}] = \frac{k_5 k_6}{k_9}\frac{P_{comS}}{\Omega} \\ \Gamma_k &= \frac{k_{-11} + k_{12}}{k_{11}} & \delta_k &= k_{12} M_T \frac{k_{11}}{k_{-11} + k_{12}} & \lambda_k &= k_8 \\ \Gamma_s &= \frac{k_{-13} + k_{14}}{k_{13}} & \delta_s &= k_{14} M_T \frac{k_{13}}{k_{-13} + k_{14}} & \lambda_s &= k_{10} \end{aligned} \tag{2.27}$$

2.2.4 Phase-plane Analysis

In order to analyze the dynamical behaviour of the system, we first look at the location of nullclines by setting $\frac{dK}{dt} = 0$ and $\frac{dS}{dt} = 0$ (The values of dimensionless parameters used in the text are given in Table 2.4, together with the original, unscaled parameters given in Table 2.3). From (2.25), the solution of the equation $\frac{dK}{dt} = 0$ is:

$$S = \frac{K}{a_k + \frac{b_k K^n}{{k_0}^n + K^n} - \Delta_k K} - K - 1 = h(K) \tag{2.28}$$

To sovle $\frac{dS}{dt} = 0$, let $u(K) = a_s + \frac{b_s}{1 + \left(\frac{K}{k_1}\right)^p}$, we have:

$$\begin{aligned} & u(K) - \frac{S}{1 + K + S} - \Delta_s S = 0 \\ & \Rightarrow \Delta_s S(K + S + 1) + S - u(K)(K + S + 1) = 0 \\ & \Rightarrow \Delta_s S^2 + S(\Delta_s K + \Delta_s + 1 - u(K)) + u(K)(K + 1) = 0 \end{aligned} \tag{2.29}$$

The last equation is quadratic, solving this equation gives us the following expression of S:

$$\begin{aligned} S &= \frac{\sqrt{\Delta_s(K+1)^2 + (u(K)-1)^2 + 2\Delta_s(K+1)(u(K)+1)} - (\Delta_s(K+1) - u(K) + 1)}{2\Delta_s} \\ &= q(K) \end{aligned}$$

α_k	$0.00875\ molec/s$	δ_k	$0.001\ s^{-1}$
α_s	$0\ molec/s$	δ_s	$0.001\ s^{-1}$
β_k	$7.5\ molec/s$	λ_k	$10^{-4}\ s^{-1}$
β_s	$0.06\ molec/s$	λ_s	$10^{-4}\ s^{-1}$

TABLE 2.3: Original, unscaled parameters of the continuous model.

a_k	0.00035	k_0	0.2
a_s	0	k_1	1/30
b_k	0.3	Δ_k	0.1
b_s	3.0	Δ_s	0.1

TABLE 2.4: Dimensionless parameters of the continuous model.

In order to find the fixed points which are located where the two nullclines intersect, we put (2.28) back into (2.29), we obtain $u(K) - \frac{h(K)}{1+K+h(K)} - \Delta_s h(K) = 0$. This equation can be numerically solved using bisection method.

Figure 2.3 shows the location of nullclines $\frac{dK}{dt} = 0$ and $\frac{dS}{dt} = 0$ on the logarithmic scaled plane of ComK and ComS together with vector field which describes the direction of trajectories. The dotted line represents the nullcline of ComS and the dashed line shows the nullcline of ComK. As we can see in the figure, there are three fixed points including the stable (full circle), saddle (empty circle) and unstable focus (rectangle) fixed points.

The stability (*stable/unstable*) of these fixed points can be inferred from their linearisation. Particularly, a fixed point is stable if all eigenvalues of its Jacobian matrix are real or complex numbers with real parts less than zero. On the contrary, it is unstable if there is at least one eigenvalue of them with real part greater than zero. From equation (2.25), the Jacobian matrix for our system has the following form:

$$\mathbf{J} = \begin{matrix} \\ (K) \\ (S) \end{matrix} \begin{matrix} \begin{matrix} (K) \end{matrix} & \begin{matrix} (S) \end{matrix} \\ \begin{pmatrix} -\Delta_k + \frac{b_k K^{n-1} {k_0}^n n}{(K^n + {k_0}^n)^2} - \frac{S+1}{(1+K+S)^2} & \frac{K}{(1+K+S)^2} \\ -\frac{b_s \left(\frac{K}{k_1}\right)^p p}{K\left(1+\left(\frac{K}{k_1}\right)^p\right)^2} + \frac{S}{(1+K+S)^2} & -\Delta_s - \frac{K+1}{(1+K+S)^2} \end{pmatrix} \end{matrix}$$

By numerically solving for the fixed points, we yield three following fixed points:

$$(K, S) = \{(0.0028, 20.488), (0.0173, 20.4880), (0.0348, 5.0851)\}$$

Thus, the corresponding eigenvalues for the left-most, intermediate and right-most fixed points as shown on the plane are:

$$(e_k, e_s) = \{(-376.5492, -367.6176), (373.3056, -348.1150),$$

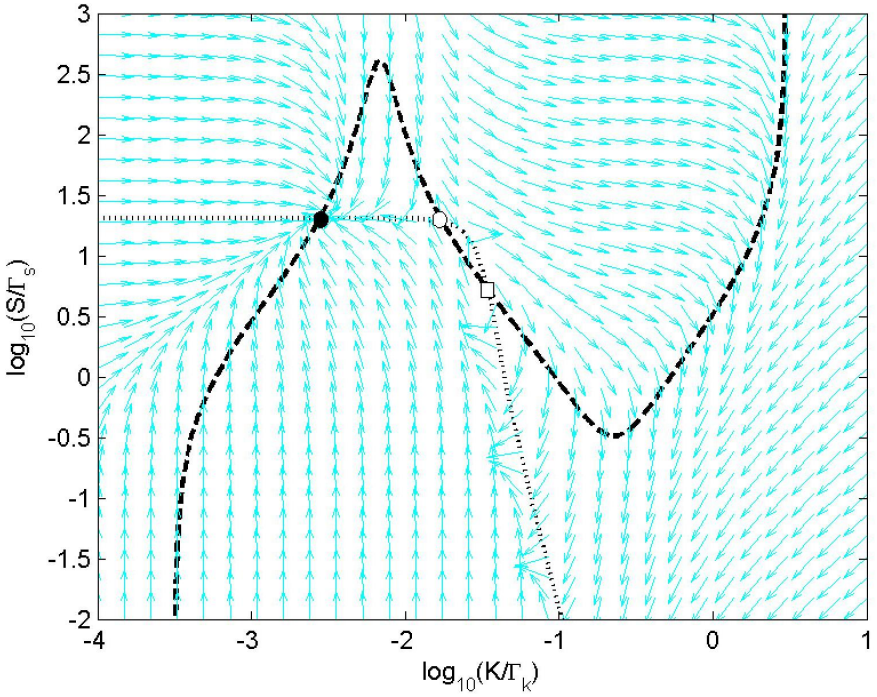


FIGURE 2.3: Nullclines and vector field of competence circuit in the continuous model.

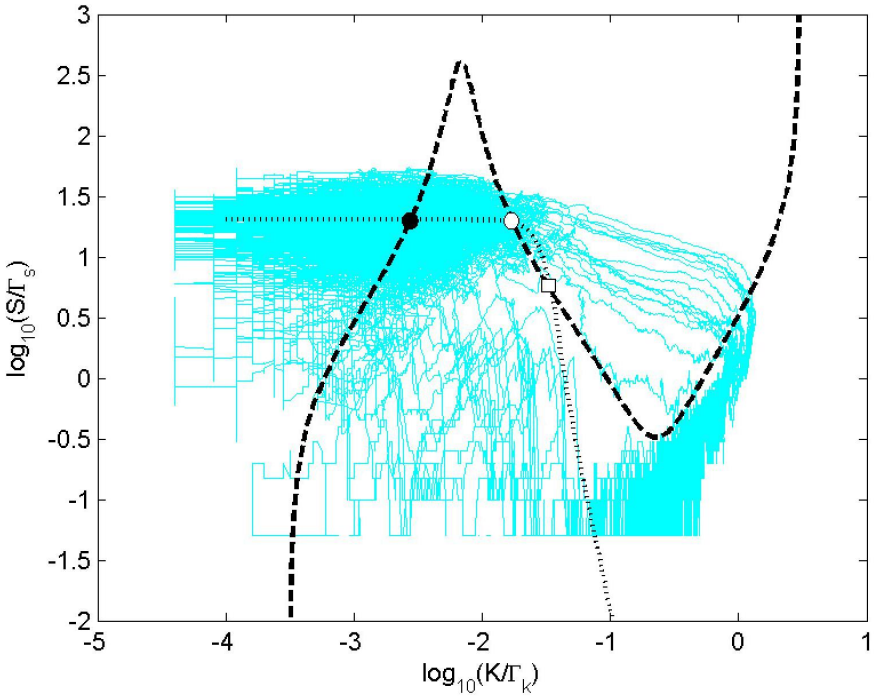


FIGURE 2.4: Structure of fixed points.

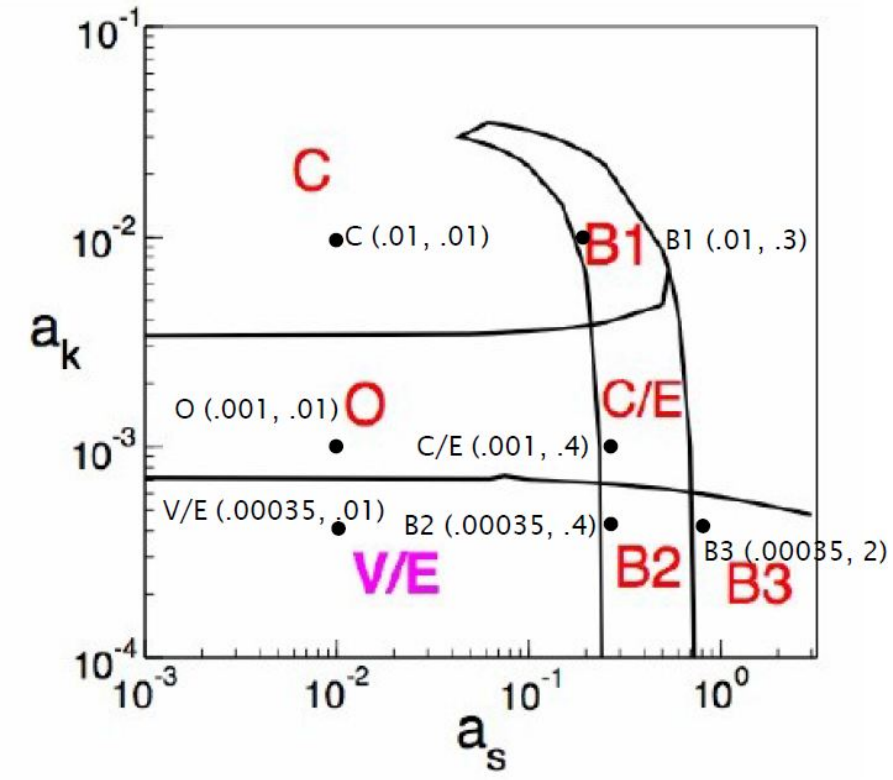
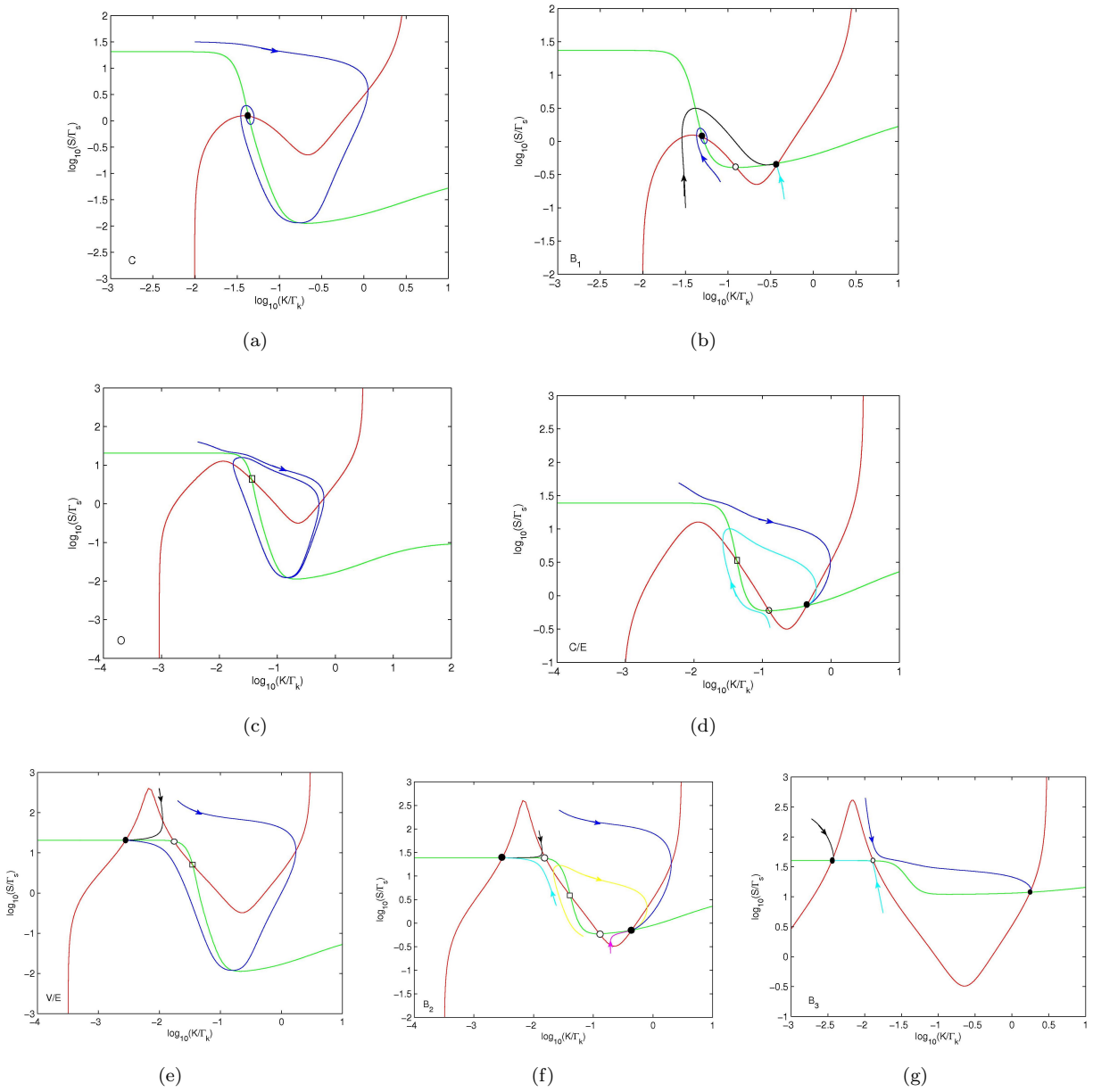


FIGURE 2.5: Diagram of different dynamical regimes (Source from Süel et al. (2007)). The dots show representative points (a_k, a_s) in each regime. Trajectories for these systems are shown in Figure 2.6.

$$183.01 + 931.82i, 183.01 - 931.82i)\}$$

As a result, the left-most fixed point is classified as stable while the intermediate and left-most fixed points are saddle and unstable focus fixed points, respectively. Figure 2.4 shows the structure of the fixed points and the trajectories generated from the discrete model (the thine line). As we can see, the dense area around the fixed point is where cells spend most of the time. As a result, it is more likely for cells to stay at the steady state (near the stable fixed point) rather than getting away from this state and come into competence where the ComK concentration is very high. In fact, the cell's behaviour can be analysed in terms of different dynamical regimes in which the system may reside (Figure 2.5). Those regimes can be analysed in the plane including V/E (Vegetative/Excitable), O (Oscillatory), C (stable competence), B1 (coexistence of two competent states with different high level of ComK), C/E ("inverse" excitable) where the rest of state is competent, B2 and B3 (coexistence of vegetative and competent states) which are different in the number of unstable states.

In order to visually demonstrate the dynamical characteristics of system at these regimes, Figure 2.6 shows sample trajectories which are created by the deterministic model given by (2.25) with the corresponding values of the model parameters for different regimes.



(a) (b) (c) (d) (e) (f) (g)

FIGURE 2.6: Phase plane plots of dynamical regimes exhibited by the model (V/E (Vegetative/Excitable) (e), O (Oscillatory) (c), C (Competence) (a), C/E ("inverse" excitable) (d), B1 (two competent states) (b), B2 (vegetative and competent states) (f), B3 (vegetative and competent states) (g)). The red (green lines) represent ComK (ComS) nullclines, the other lines denote sample trajectories. Stable fixed points are denoted as full circles, saddle points as empty circles and other unstable points as rectangles.

I used the *ode*45 solver (the ordinary differential equation solver) in Matlab to compute the trajectories for each set of parameters, the systems are shown by the points marked in Figure 2.5. In Figure 2.6, starting from the V/E regime, we keep a_s unchanged but gradually increase the value of a_k. In this case, the nullcline of ComS remains the same but the nullcline of ComK starts moving downward. As the peak of the ComK nullcline lies below the ComS nullcline, crossing the boundary between V/E and O states (Figure 2.5), then the two fixed points which are denoted as circles disappear leaving an unstable point. Consequently, there is a transition in the cellular behaviour from the V/E (Vegetative/Excitable) state to the O (Oscillatory) state. Likewise, cells may change from the oscillatory state to the competent state upon increasing. This makes the nullcline of ComK also move downward and therefore, the unstable point becomes a stable point as can be seen in the figure. We notice that the wild type is believed to be in state V/E. From now on, we will therefore assume we are in the V/E state.

2.2.5 Probability of Initiation and Competence Duration

In Süel et al. (2007), the authors mentioned about the probability of initiation of competence, P_{init}, which is defined as the probability per cell cycle that a cell becomes competent. In order to estimate P_{init}, they suggested the following way: For a fixed number of realizations c ($c = 10$ in our simulation) leading to competent events, we compute the time needed for the initiation of competent events to occur. Assuming a cell-cycle time of 4 hours, we calculate the number of cell divisions that have occurred until competence arises by taking a sum of the inter-competence event durations which are time durations between two consecutive competent events (see Figure 2.7) divided by cell-cycle time. Dividing the total number of competent events by the number of cell divisions gives P_{init}. The competence duration, τ_{comp} is computed as the time during which the ComK molecules exceeds a threshold which is taken to be 10^4. I will show the calculation result of reproducing how the initialization probability varies by changing the model parameters by 20%. The changes in the probability of initiation of competence as well as the competence duration are illustrated in Figure 2.8. As we can see, the initialization probability as well as the competence duration change quite significantly when changing $b_k, \Delta_k, \Delta_s, k_k$ while it does not change much after varying k_s. This means the dynamical behaviour is less sensitive to the variation of k_s compared to the other parameters. In our simulation, however, the competence duration roughly varies around 10 hours compared to 20 hours mentioned in the Supporting Online Material in Süel et al. (2007) (Figure 2.9). This discrepancy comes from the fact that their simulation results were inconsistent. In particular, when reconstructing the competence circuit by introducing an additional transcription negative feedback loop onto *comK*, forming CompRok strain, they estimated the competence duration using fluorescence time traces and compared with that in the wild-type. Even though they claimed the competence duration in the wild-type was 20.2 ± 9.9 hours (mean $\pm$ SD) compared to

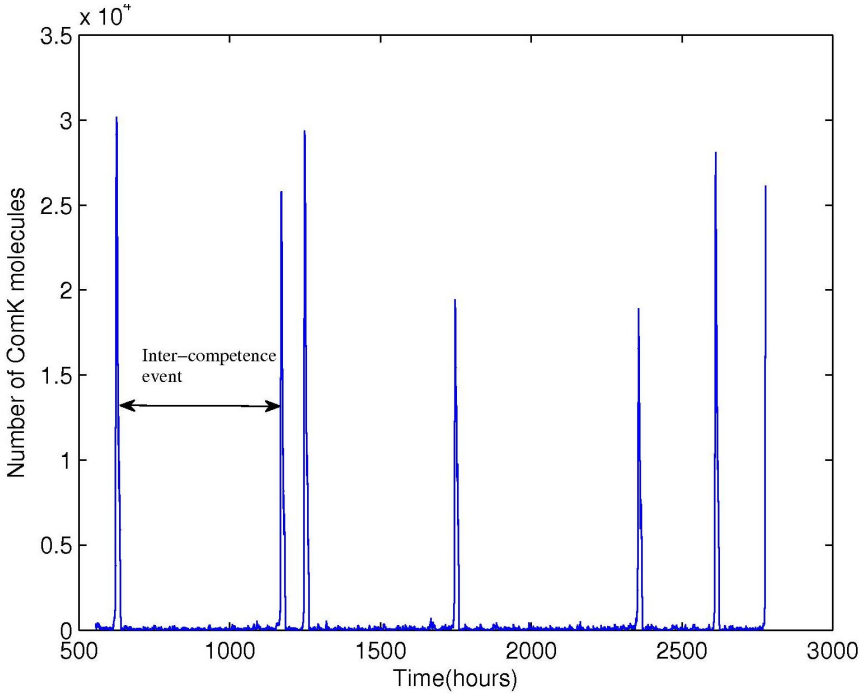


FIGURE 2.7: Inter-competence event duration

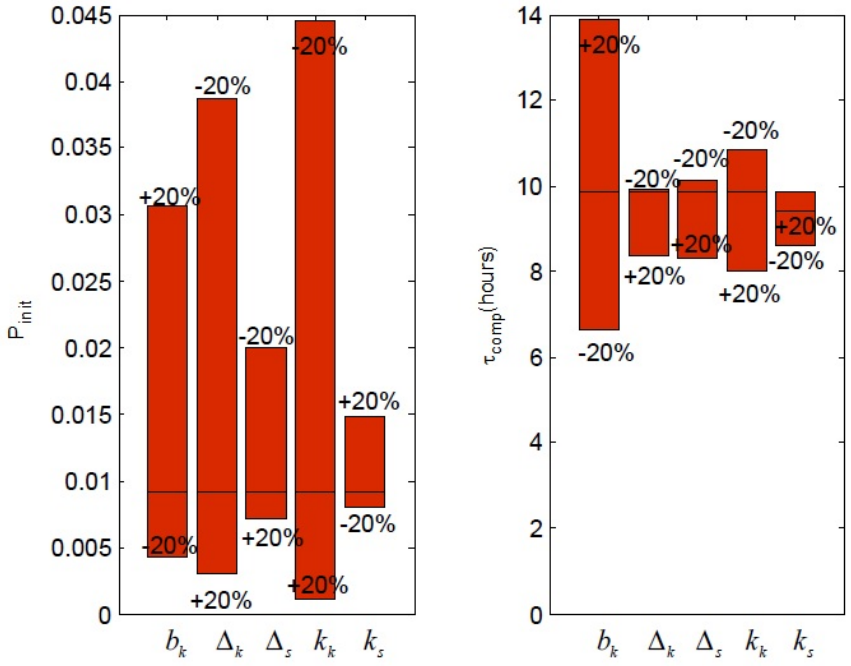


FIGURE 2.8: The probability of initiation (left) and competence duration (right) as increasing and decreasing the values of parameters of model by 20%.

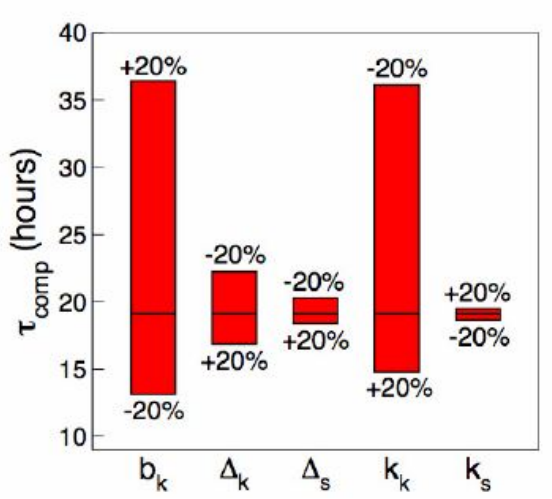
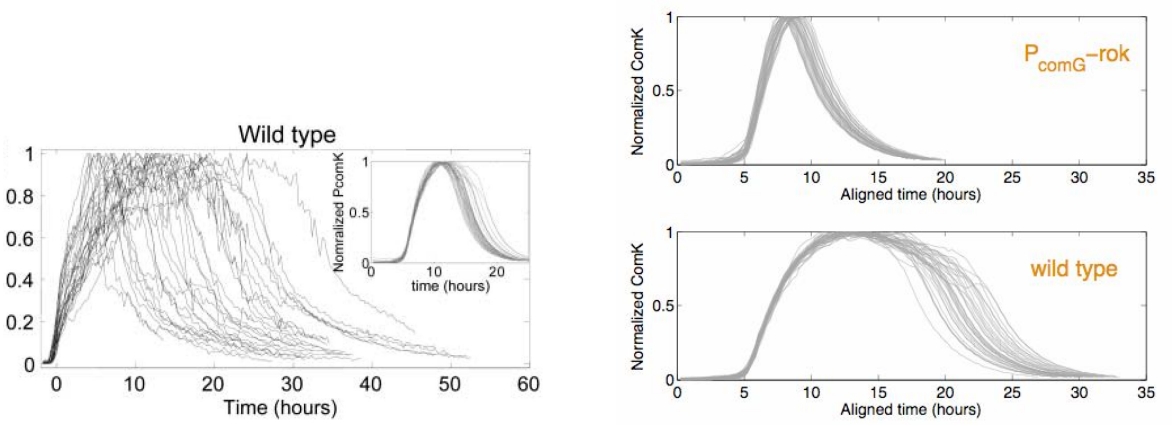
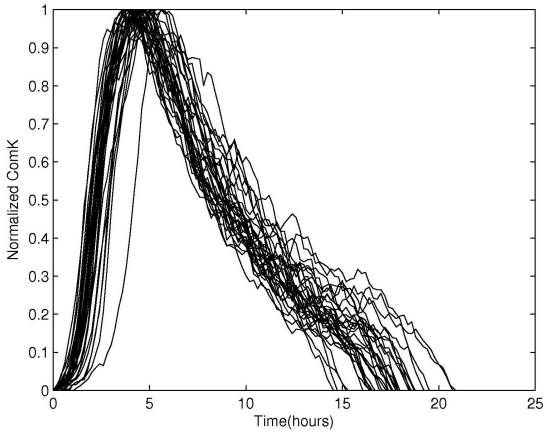


FIGURE 2.9: Competence durations with different sets of model parameters (source from Süel et al. (2007)).



(a) (b)



(c)

FIGURE 2.10: Normalized P_{comK} in the discrete stochastic simulations in wild-type competence circuit (a), which is inconsistent with the normalized ComK (b) (source from Süel et al. (2007)), and the competent events normalized by the maximum value of ComK in our simulation (c), in which the competence duration is just about 10 hours which agrees with that shown in Figure (a).

13.9 ± 3.4 hours (mean $\pm$ SD) in the CompRok strain, it was inconsistent with their simulation result mentioned in the main text (see Figure 2.10) where the competence duration was just about 10 hours. On the other hand, the figure showed the normalized quantity of P_{comK} while P_{comK} is supposed to be constant in their model. Moreover, this quantity is inconsistent with the simulation result showed in the Supporting Online Material they provided. In order to verify this, I did the discrete stochastic simulation for the wild-type circuit using Dizzy with parameters given in the paper (the Dizzy file can be found in Appendix A.11). The simulation was done for 10000 hours with 33 ($c = 33$) competent events being observed, the competence duration was computed to be $11.0 \pm \frac{1.2}{\sqrt{33}}$ hours (mean $\pm \frac{SD}{\sqrt{c}}$, $c = 33$).

2.3 Summary

In this chapter, I have reproduced some simulation results of the wild-type genetic circuit which have been given in Süel et al. (2007). In doing so, we have conducted an analysis to show how the cellular behaviour of the system can be divided into different dynamical regimes. This is very important since it helps understand how cells behave differently in each particular regimes, and suggests experimental interventions that can probe the properties of biological systems that are in the neighbourhood of the particular system under study. In the simulation, I also showed an inconsistency when computing competence duration mentioned in Süel et al. (2007). Particularly, this is not only the different competence durations observed but also the difference in the sensitivity of the results obtained. This suggests that the authors might have measured different quantity rather than competence duration. In addition, our simulations also showed how the initialization probability and competence duration depend on the variation of the model parameters. As a result, it describes how sensitive those parameters to the excitable regimes.

In the next chapter, I will provide a deeper insight into the weakness of the model in which I show that the existing model is not good enough to describe the dynamical behaviour of the system in terms of the competence duration in the excitable state. To overcome this, I propose solutions to the problem in order to simplify the system and at the same time capture the right dynamics of the system in that particular excitable state.

Chapter 3

Reduction of Deterministic Model in Competence Regime

In this chapter, I will address the inaccuracy in describing the competence duration of the system during the excitable state in Süel et al. (2007). In particular, the authors tried to reduce the full system to a 2D system given by equations (2.24) and (2.25) by applying the adiabatic approximation for the fast processes which do not potentially contribute to the dynamics of the system; and therefore, can be eliminated. However, as I will show, this approach provides a quantitatively poor approximation of the model in the excitable regime where cells are in the competent state. In order to solve the problem, I present a method which is a combination of using the adiabatic approximation and slow-invariant manifold method to obtain a better deterministic model for that particular regime.

3.1 Discrepancy in Competence Duration

The 2D adiabatic model given by equation (2.25) is reduced from the full system by using adiabatic approximation for the mRNA and the complex protease. However, the assumption produces an inconsistency in timescale between this model and the full 7D model from which it has been reduced. In this scenario, we do the simulation for the excitable regime where the competent events occur and make a comparison between the 2D adiabatic model and the 7D model. For simplicity, we denote the species concentrations $[MecA]$, $[MecA_K]$, $[MecA_S]$, $[ComK]$, $[ComS]$, $mRNA_{comK}$, $mRNA_{comS}$ as A, M_k, M_s, K, S, R_K, R_S. The 7D deterministic model can be obtained from the full stochastic system by taking the limit of the system volume to infinity as discussed in section 1.1.3. The differential equations for a 7D system are re-written as

below:

$$\begin{aligned}
\frac{dR_K}{dt} &= k_1 + \frac{k_2 K^n}{{k_k}^n + K^n} - k_7 R_K \\
\frac{dR_S}{dt} &= k_4 + \frac{k_5}{1 + (K/k_s)^p} - k_9 R_S \\
\frac{dA}{dt} &= -k_{11} K A + k_{-11} M_K + k_{12} M_K + k_{14} M_S \\
&\quad - k_{13} S A + k_{-13} M_S \\
\frac{dM_K}{dt} &= -k_{12} M_K + k_{11} K A - k_{-11} M_K \\
\frac{dM_S}{dt} &= -k_{14} M_S + k_{13} S A - k_{-13} M_S \\
\frac{dK}{dt} &= -k_{11} K A + k_{-11} M_K + k_3 R_K - k_8 K \\
\frac{dS}{dt} &= -k_{13} S A + k_{-13} M_S + + k_6 R_S - k_{10} S
\end{aligned} \tag{3.1}$$

In the simulation, I use the Matlab solver *ode45* to generate trajectories for each deterministic models for the comparison, and the numerical initial condition for the integration is $K = 1099$, $S = 564$, $A = 16$, $M_K = 3$, $M_S = 481$, $R_K = 1$, $R_S = 0$.

The result of simulation shows that, in the 2D adiabatic model, cells spend about 3.9 hours in the competent state before coming back to the vegetative state. In the 7D system, however, cells reside in competent state for approximately 10.1 hours (Figure 3.1). In addition, the trajectory created by the 7D deterministic model (3.1) does not follow the vector field of the 2D adiabatic model (2.24) when coming back to the vegetative state (Figure 3.2). In fact, the cooperative binding/unbinding of protein to the complex protease are not fast processes as assumed ($k_{11} = 2.02 \times 10^{-6}$, $k_{-11} = 5 \times 10^{-4}$, $k_{13} = 4.5 \times 10^{-6}$, $k_{-13} = 5 \times 10^{-5}$). In other words, it is a slow process of binding and unbinding among protein and the complex protease that makes the degradation of protein due to being absorbed by the protease slow; therefore, the time that the cells spend in competent state lasts longer.

To verify our hypothesis, we speed up those processes by increasing the reaction rates including k_{11}, k_{-11}, k_{12}, k_{13}, k_{-13}, k_{14} by a factor of $r = 10$ and at the same time reducing the total concentration of MecA by the same factor so that the speed-up does not effect the parameters of the 2D adiabatic model (see Equation (2.27) in Chapter 2). The experimental result shows that the trajectory follows the vector field (Figure 3.3) and the competence duration is much closer to that in the 2D adiabatic model. In particular, the time series of ComK are almost the same in the 7D and 2D models when the chosen rate constants are sped up by a factor $r = 100$ (Figure 3.4).

In conclusion, the 2D adiabatic model is inaccurate for modeling the dynamics of the system at the excitable regime where the competence occurs. In particular, the competence

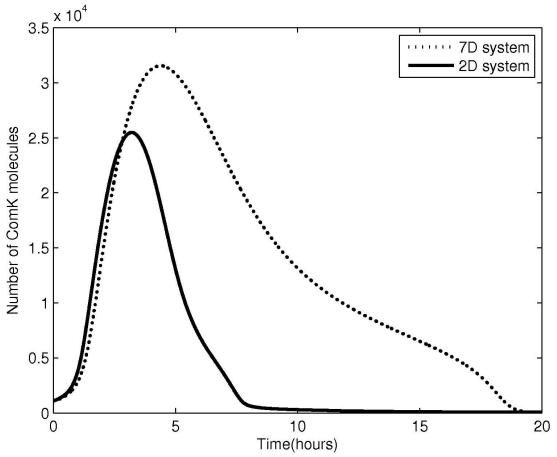


FIGURE 3.1: Time series of ComK.

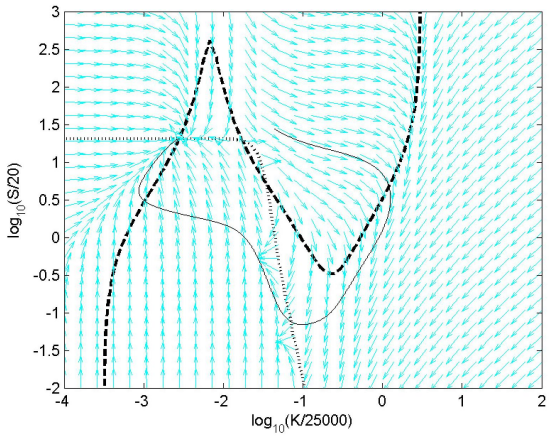


FIGURE 3.2: Trajectory created by the 7D deterministic model.

duration is shorter than that in the full system by a factor of 2.5. For this reason, we need to find a solution to the problem of dimensionality reduction which allows to simplify our full system to lower-dimensional system while still preserving the competence duration. This solution will be systematically built up in the following section.

3.2 Model Reduction

The problem of dimensionality reduction is critical in researching dynamical systems. The main aim of this is to reduce a very high-dimensional system to lower-dimensional system so that the system analysis can be done more easily. The solution to this problem has been developed by a number of authors. In this section, however, we are going to study a relatively efficient method called the multi-scale technique which has been widely used in stochastic simulation, especially in chemical reaction systems (Bennett et al., 2007; Lee and Othmer, 2010; Cotter et al., 2011). The main idea of this method is to capture fast and slow species in the system based on the reactions they participate in. In particular, all species which get changed by fast reactions are seen as fast species.

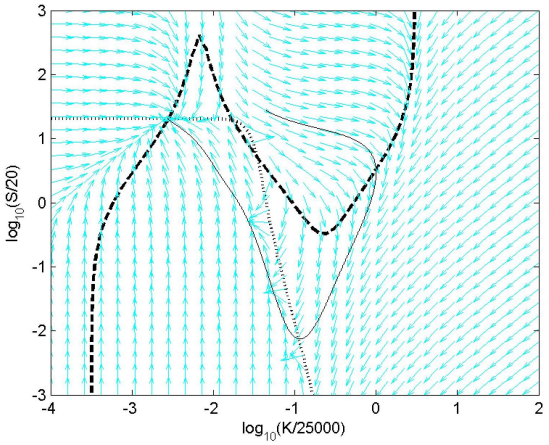


FIGURE 3.3: Trajectory created by the 7D deterministic model after the speed-up.

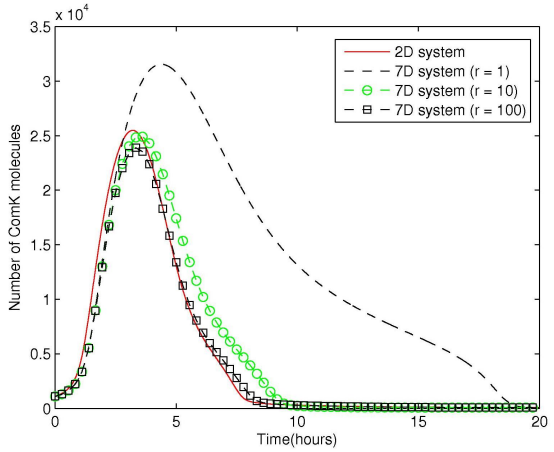


FIGURE 3.4: Comparison in competence duration between the 7D and 2D adiabatic models after speed-up.

Otherwise, they are defined to be slow species. In fact, those species can be characterized by computing the rates of change in their concentrations during the time evolution of the system. From the perspective of the simulation system, the slow species are invariant with respect to the fast reactions on a fast time scale meanwhile the fast species quickly reach equilibrium on a slow time scale. As a result, the problem of reduction can be solved by setting fast and slow species to their starting values and quasi-steady states, respectively. However, the characteristic of these species may not be defined if they occur in both fast and slow reactions; therefore, the multi-scale method in general does not seem to be a good choice for solving the problem. To overcome this, Bennett et al. (2007) suggested a multiple time-scale approach in which all species are partitioned into different parts, then forming a hierarchical *slow invariant manifold* evolving on different time scales. Consequently, each part which attracts the flow of the system contains a lower-dimensional slow manifold which can be identified for reducing the dimensionality of the full-dimensional system. This method, however, is quite complicated when dealing with a sophisticated high-dimensional system with too many variables. In our system, we are going to try to address the problem by capturing the slow invariant manifold

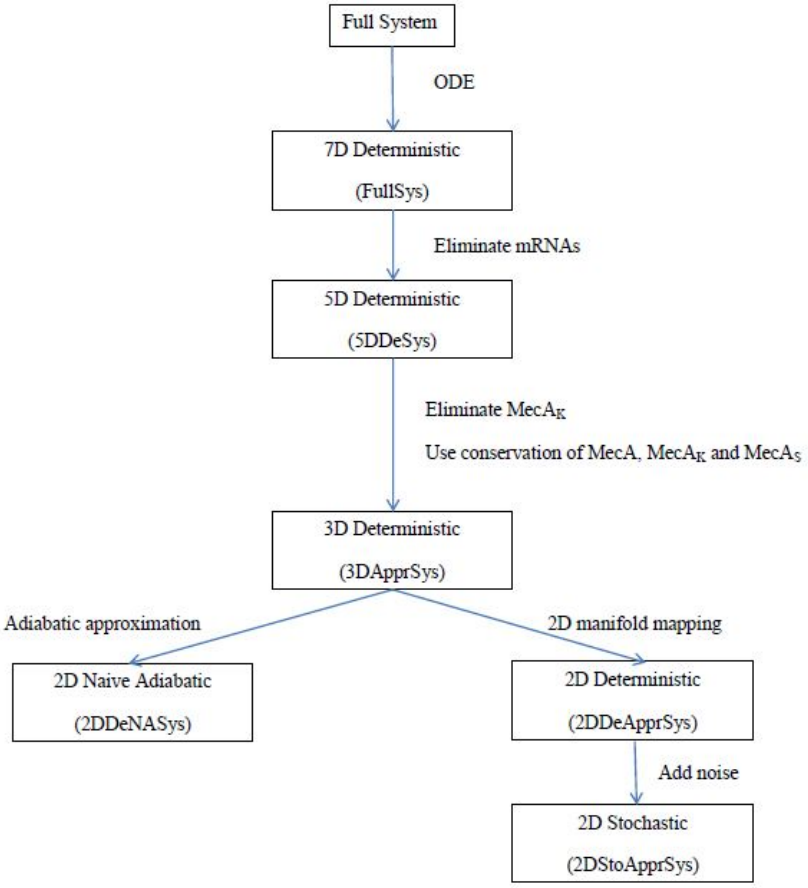


FIGURE 3.5: Structure of different models. The arrows shows an approximation of high-dimensional system to a lower-dimensional system.

in a simpler way which is a combination of the slow invariant manifold method and an iterative procedure suggested by Roussel and Fraser (Roussel and Fraser, 1990, 2001).

In the following sections, we will try to reduce our system to lower-dimensional systems and make a comparison with other systems in terms of trajectories, competence duration, etc. Our purpose is to come up with a reduced model which can best approximates the cellular behaviour after the system enters the competent state. There are two critical properties of the system dynamics we are interested in: the first one is the competence duration; the second one is the stationary probability distribution which describes the probability of the system being at a particular state. In this chapter, we focus on the first property and try to find the approximate model which best preserves the competence duration. Figure 3.5 shows the different systems we are going to work on including the full discrete stochastic model (Full System), the 7D deterministic system (FullSys), the 5D deterministic system (5DDeSys), the 3D deterministic model (3DApprSys), the 2D naive adiabatic model (2DDeNASys) and the 2D deterministic model (2DDeApprSys).

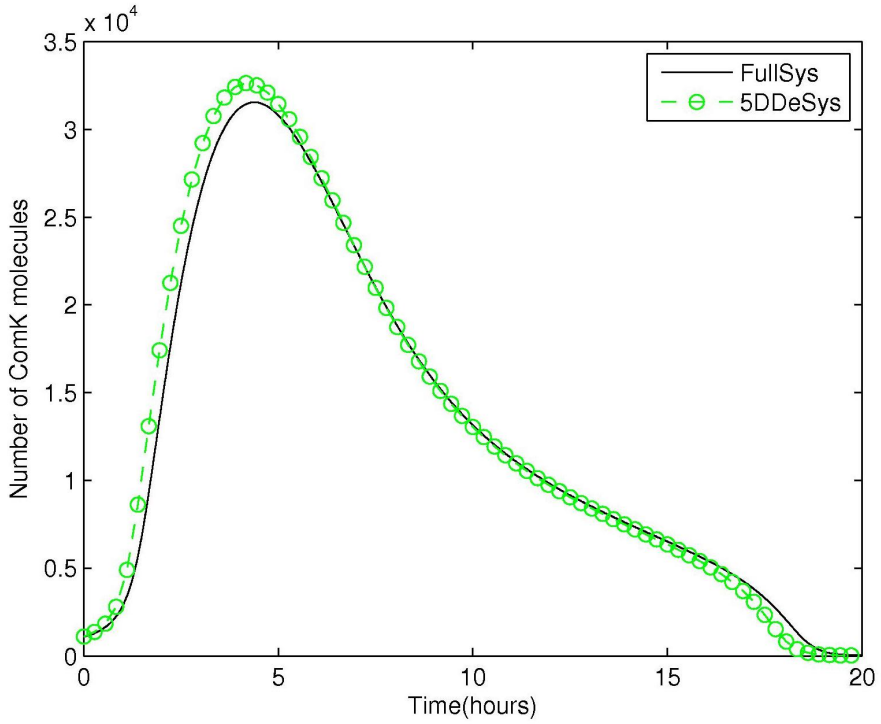


FIGURE 3.6: Time series of ComK in the 7D (FullSys) and 5D deterministic (5DDeSys) models.

3.2.1 Five-dimensional Deterministic System (5DDeSys)

The key idea of dimensionality reduction problem is to identify fast and slow reactions or conservative variables. Since the model is in a high expression state for ComK, we note that the higher decay rates of mRNA ($k_7 = k_9 = 0.005$) compared with proteins ($k_8 = k_{10} = 10^{-4}$) can be used to justify the treatment of mRNA as fast variables; therefore, we can eliminate the mRNAs by setting them to their steady state values ($dR_{K,S}/dt = 0$). Consequently, the full model described in (3.1) can now be reduced to a 5D deterministic model (5DDeSys) as follows:

$$\frac{dK}{dt} = -k_{11}KA + k_{-11}M_K + \frac{(k_3k_1 + \frac{k_2k_3K^n}{k_k{}^n + K^n})}{k_7} - k_8K \tag{3.2}$$

$$\frac{dS}{dt} = -k_{13}SA + k_{-13}M_S + \frac{(k_4k_6 + \frac{k_5k_6}{1+(\frac{K}{k_s})^p})}{k_9} - k_{10}S \tag{3.3}$$

$$\frac{dA}{dt} = -k_{11}KA + k_{-11}M_K + k_{12}M_K + k_{14}M_S - k_{13}SA + k_{-13}M_S \tag{3.4}$$

$$\frac{dM_K}{dt} = -k_{12}M_K + k_{11}KA - k_{-11}M_K \tag{3.5}$$

$$\frac{dM_S}{dt} = -k_{14}M_S + k_{13}SA - k_{-13}M_S \tag{3.6}$$

In order to test if the 5D deterministic model preserves the competence duration, we do the simulation for the 7D (FullSys) and 5D deterministic (5DDeSys) models using the Matlab solver *ode45*, then compare the time series of ComK for both models. The simulation result shows that the competence duration computed in both model is around 10.1 hours (Figure 3.6). This means we successfully reduce the full model to a 5D model while still preserving the competence duration. In the following section, we will try to reduce the 5D model to a 3D model.

3.2.2 Three-dimensional Approximate System (3DApprSys)

Let us consider the following reactions:

$$\begin{aligned} \text{MecA} + \text{ComK} &\xrightarrow{k_{11}/\Omega} \text{MecA}_\text{K} \\ \text{MecA}_\text{K} &\xrightarrow{k_{-11}} \text{MecA} + \text{ComK} \\ \text{MecA}_\text{K} &\xrightarrow{k_{12}} \text{MecA} \\ \text{MecA} + \text{ComS} &\xrightarrow{k_{13}/\Omega} \text{MecA}_\text{S} \\ \text{MecA}_\text{S} &\xrightarrow{k_{-13}} \text{MecA} + \text{ComS} \\ \text{MecA}_\text{S} &\xrightarrow{k_{14}} \text{MecA} \end{aligned} \tag{3.7}$$

Since $k_{12} \gg k_{11}, k_{-11}$, therefore the degradation process of $MecA_K$ is much faster than the others. In other words, $MecA_K$ will quickly reach its steady state at which $\frac{dM_K}{dt} = 0$. As a result, we have $M_K = \frac{k_{11}}{k_{12}+k_{-11}} KA = \frac{KA}{\Gamma_k}$. Using the conservation equation $M_T = A + M_K + M_S$, we obtain (for $Q = M_T - M_S$)

$$M_K = \frac{(K/\Gamma_k)}{1 + (K/\Gamma_k)} Q$$

We introduce dimensionless variables, $K \mapsto (K/\Gamma_k)$, $S \mapsto (S/\Gamma_s)$, $Q \mapsto (Q/M_T)$ (which ranges from 0 to 1), in terms of which we obtain the following differential equations (Notice that $k_4 = 0$):

$$\frac{dK}{dt} = \frac{k_3}{\Gamma_k k_7} \left(k_1 + \frac{k_2 K^n}{(k_k/\Gamma_k)^n + K^n} \right) - \frac{k_{12} M_T}{\Gamma_k} \frac{KQ}{1+K} - k_8 K \tag{3.8}$$

$$\frac{dS}{dt} = \left(\frac{(k_5 k_6)/(\Gamma_s k_9)}{1 + (\Gamma_k/k_s)^p K^p} \right) - k_{10} S - \frac{k_{14}}{\Gamma_s} M_T (1 - Q) + k_{13} M_T \left(1 - \left(1 + \frac{S}{1+K} \right) Q \right) \tag{3.9}$$

$$\frac{dQ}{dt} = \Gamma_s k_{13} \left(1 - \left(1 + \frac{S}{1+K} \right) \right) \tag{3.10}$$

Upon introducing new parameters to simplify the appearance of these equations, as before, we arrive at

$$\begin{aligned}
\frac{dK}{dt} &= a_k + \frac{b_k K^n}{k_0^n + K^n} - \frac{KQ}{1+K} - \Delta_k K \\
\frac{dS}{dt} &= \frac{d_s}{1+(K/c_s)^p} - \mu(1-Q) + \epsilon_s \left(1 - \left(1 + \frac{S}{1+K}\right) Q\right) - \Delta_s S \\
\frac{dQ}{dt} &= \epsilon_q \left(1 - \left(1 + \frac{S}{1+K}\right) Q\right)
\end{aligned} \tag{3.11}$$

with the rescaled time variable $t \mapsto (k_{12} M_T/\Gamma_k)t$. The parameters are defined with respect to those in the CME as follows:

$$\begin{aligned}
\Gamma_k = \tfrac{k_{-11}+k_{12}}{k_{11}}, \quad \Gamma_s = \tfrac{k_{-13}+k_{14}}{k_{13}}, \quad a_k = \tfrac{k_3 k_1}{k_7 k_{12} M_T}, \quad \Delta_k = \tfrac{\Gamma_k k_8}{k_{12} M_T} \\
k_0 = \frac{k_k}{\Gamma_k}, \quad c_s = \frac{k_s}{\Gamma_k}, \quad d_s = \frac{k_5 k_6 \Gamma_k}{k_9 k_{12} M_T \Gamma_s}, \quad \Delta_s = \frac{\Gamma_k k_{10}}{k_{12} M_T} \\
\epsilon_s = \frac{k_{13}\Gamma_k}{k_{12}}, \quad \epsilon_q = \frac{\Gamma_k \Gamma_s k_{13}}{k_{12} M_T}, \quad \mu = \frac{k_{14}\Gamma_k}{\Gamma_s k_{12}}.
\end{aligned} \tag{3.12}$$

The system can now be described in the 3D deterministic model (3DApprSys) given by equation 3.11. Accordingly, the trajectories can now be computed and compared with that in the 7D deterministic model (FullSys), Figure 3.7 shows that those trajectories perfectly match at the excitable regime. Furthermore, the competence duration computed in the 3D deterministic model is the same as that in the 7D deterministic model (around 10.1 hours) (Figure 3.8). As a result, the 5D deterministic model can be replaced with a 3D deterministic model which is much easier for analysis as well as computation afterwards. In the previous section, Süel et al. (2007) introduced a 2D naive adiabatic model which is simple for modelling the dynamics of cell's behaviour. Unfortunately, we have found that this model does not describe the dynamics of system as expected. In particular, the observed competence duration is off by a factor of 3. In the following section, we are trying to approximate the full system by a 2D approximate system and at the same time, compare with the 2D naive adiabatic system in terms of trajectory as well as competence duration. Since the full model is well approximated by the 3D deterministic model, we therefore just need to compare the 2D approximate model with this 3D deterministic model.

3.2.3 A Two-dimensional Approximate System

In the 3D deterministic model, there are no obvious fast variables. However, there may be fast processes involving the interaction of several variables. To study this, we look at the eigenvalues of the Jacobian matrix computed at every single point along the average trajectory. The Jacobian matrix measures the stability of the trajectory to changes in the parameters. That is, if $\mathbf{X} = (K, S, Q)$ then we can define the dynamics by $\dot{\mathbf{X}} = f(\mathbf{X})$

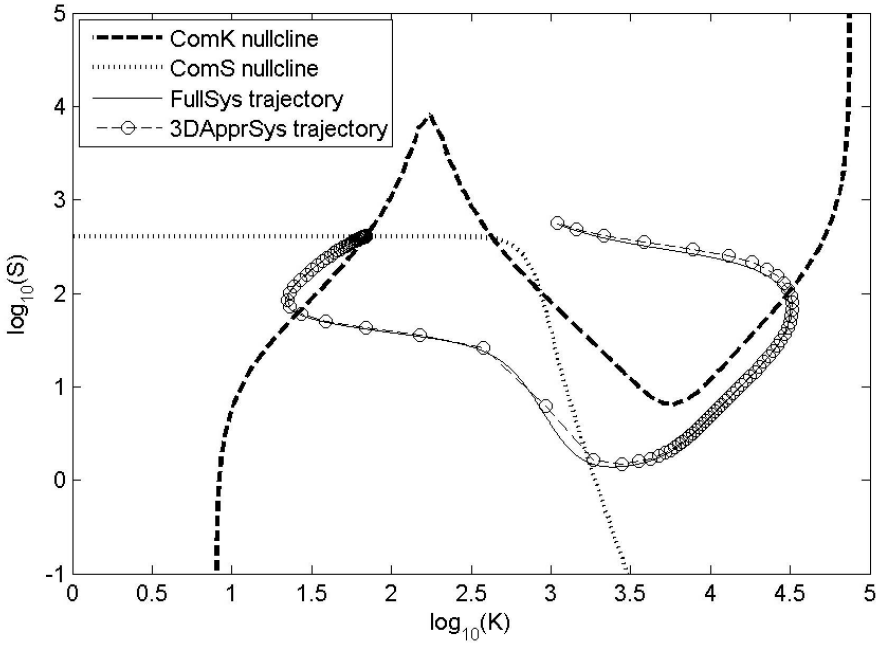


FIGURE 3.7: Trajectories in 3D (3DApprSys) and 7D deterministic (FullSys) models.

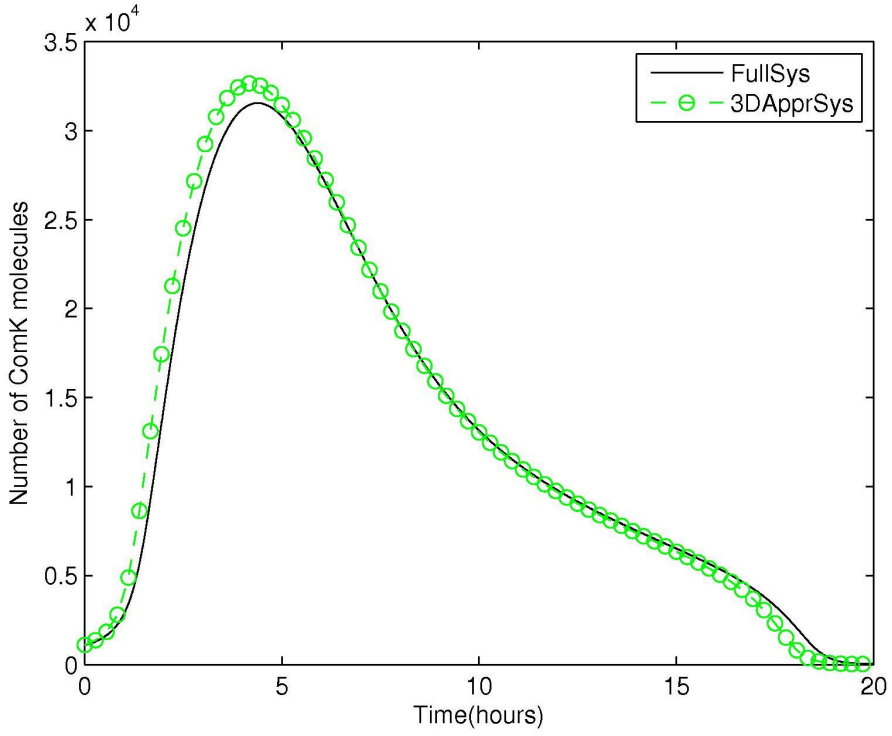


FIGURE 3.8: Time series of ComK in 7D (FullSys) and 3D deterministic (3DApprSys) models.

and we consider $\delta\dot{\mathbf{X}}(t) = f(\mathbf{X}+\delta\mathbf{X}) - f(\mathbf{X}) = \mathbf{J}\delta\mathbf{X}(t)$. Where $\mathbf{J}$ is the Jacobian matrix with components $J_{i,j} = \frac{\partial f_i(\mathbf{X})}{\partial X_j}$. As a result, the solution is $\delta\mathbf{X}(t) = \sum c_i \boldsymbol{\nu}_i e^{\lambda_i t}$, where $\boldsymbol{\nu}_i$ are the right eigenvectors corresponding to eigenvalues λ_i, and c_i are the components of $\mathbf{X}(t)$ along $\boldsymbol{\nu}_i$. Large negative eigenvalues point to the rapid decay of deviations, and a widely separated set of eigenvalues enables us to eliminate these fast decaying

modes. In more details, Figure 3.9 shows a plot of eigenvalues computed along the average trajectory for the 3D system described in (3.11) on a 10-base logarithm scaled polar coordinates.

In order to compute the average trajectory, we sample the data from the Gillespie simulation of the full system; we then choose polar coordinates such that the origin is positioned inside the region limited by all sampled trajectories. To construct the polar coordinates, we divide the space into 360 equal small sector shaped subspaces where each region i is defined by an angle $\theta_i = \frac{i\pi}{180}$, $i = 1, 2, \ldots, 360$ as can be seen in the figure. Next, all the data which falls into a specific region will be stored for calculation. Upon averaging in each of the unit-degree regions, we create an average trajectory by joining these averages. In this figure, a particular eigenvalue is plotted in such a way that the distance from it to the origin is computed by taking a 10-base logarithm of its inverse absolute value.

It is clear that the 3 eigenvalues are separated from each other during the excitable state back to the vegetative state, making possible a reduction to a lower-dimensional system. In our case, the most negative eigenvalues are about 10 times as large as the others in absolute value, implying the existence of a low-dimensional attracting manifold. However, there also exists positive eigenvalues marked in Figure 3.9 which is the hallmark of an excitable system. For this reason, the whole space is divided into subspaces which are defined by positive and negative eigenvalues; the subspaces where the positive eigenvalues are found are demarcated by angles α, β.

We have shown that, in the 3D system (K, S, Q), there is a fast relaxation mode along the whole trajectory. As a consequence, almost all trajectories will lie close to a 2D manifold. We can therefore reduce our system to a 2D system by assuming Q is a function of K and S, ie. $Q = Q(K, S)$. This means the dynamics of the system always lies close to a 2D manifold in (K, S, Q) space and its velocity is uniquely determined by K and S alone. As a result, we have $\frac{dQ}{dt} = \frac{dQ}{dK}.\frac{dK}{dt} + \frac{dQ}{dS}.\frac{dS}{dt}$. Plugging this back into equation 3.11 we obtain:

$$
\begin{aligned}
\epsilon_q \left(1 - \left(1 + \frac{S}{1+K}\right) Q\right) &= \frac{dQ}{dK}\left(a_k + \frac{b_k K^n}{k_0^n + K^n} - \frac{KQ}{1+K} - \Delta_k K\right) \\
&\quad + \frac{dQ}{dS}\left(\frac{d_s}{1 + (K/c_s)^p} - \mu(1-Q) + \epsilon_s \left(1 - \left(1 + \frac{S}{1+K}\right) Q\right) - \Delta_s S\right) \\
&= \frac{dQ}{dK}\left(a_k + \frac{b_k K^n}{k_0^n + K^n} - \Delta_k K\right) - \frac{dQ}{dK}\frac{KQ}{1+K} \\
&\quad + \frac{dQ}{dS}\left(\frac{d_s}{1 + (K/c_s)^p} - \mu + \epsilon_s - \Delta_s S\right) + \frac{dQ}{dS}\left(\mu - \epsilon_s \left(1 + \frac{S}{1+K}\right)\right) Q \\
&= \frac{dQ}{dK}\left(a_k + \frac{b_k K^n}{k_0^n + K^n} - \Delta_k K\right) + \frac{dQ}{dS}\left(\frac{d_s}{1 + (K/c_s)^p} - \mu + \epsilon_s - \Delta_s S\right) \\
&\quad - Q\left(\frac{dQ}{dK}\frac{K}{1+K} - \frac{dQ}{dS}\left(\mu - \epsilon_s \left(1 + \frac{S}{1+K}\right)\right)\right)
\end{aligned}
$$

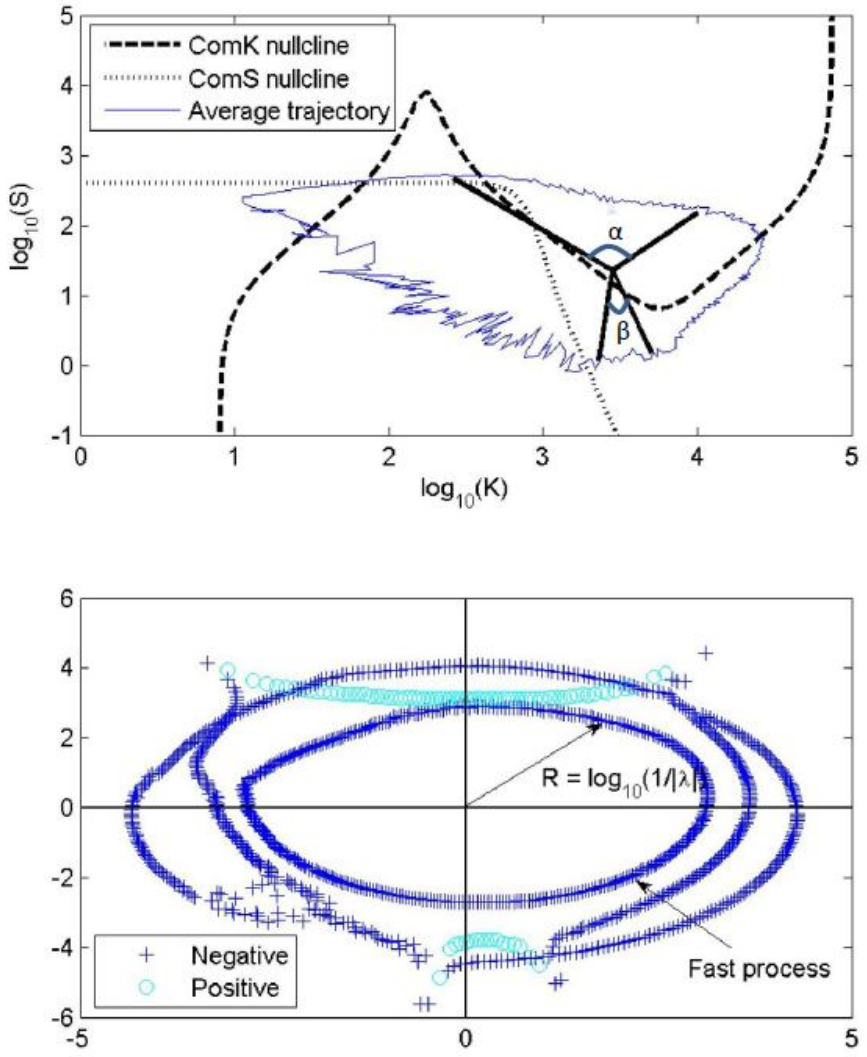


FIGURE 3.9: The spectra of eigenvalues on a 10-based logarithm scale. The position of the eigenvalue point is defined by the angle formed by the data point at which the eigenvalue is evaluated and the vertical axis of the polar coordinates, and the distance from that point to the origin. This distance is computed by taking the logarithm of the inverse absolute eigenvalue.

$$\Rightarrow Q = \frac{\epsilon_q - \frac{dQ}{dK}\left(a_k + \frac{b_k K^n}{k_0^n + K^n} - \Delta_k K\right) - \frac{dQ}{dS}\left(\frac{d_s}{1+(K/c_s)^p} - \mu + \epsilon_s - \Delta_s S\right)}{\epsilon_q - \frac{dQ}{dK}\frac{K}{1+K} + \frac{dQ}{dS}\mu + \left(1 + \frac{S}{1+K}\right)\left(\epsilon_q - \frac{dQ}{dS}\epsilon_s\right)}$$
$$= F\left(K, S, \frac{dQ}{dK}, \frac{dQ}{dS}\right)$$

In order to estimate function Q, we use an iterative procedure (Fraser, 1988; Roussel, 1997) in which by starting from an initial trial function $Q_0(K, S)$ we compute $Q_{n+1} = F(K, S, \frac{dQ_n}{dK}, \frac{dQ_n}{dS})$, $n = 1, 2, \ldots$. Choosing $Q_0 = 0$, for example, we then obtain a set of iteratively defined functions Q_n as follows (The numerical values of parameters are

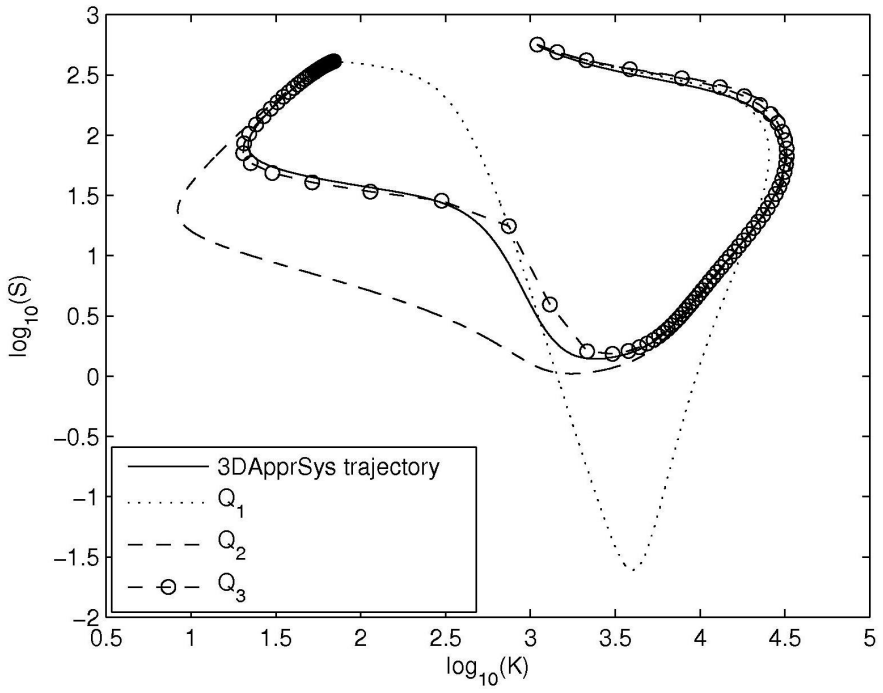


FIGURE 3.10: Trajectories created by different deterministic models.

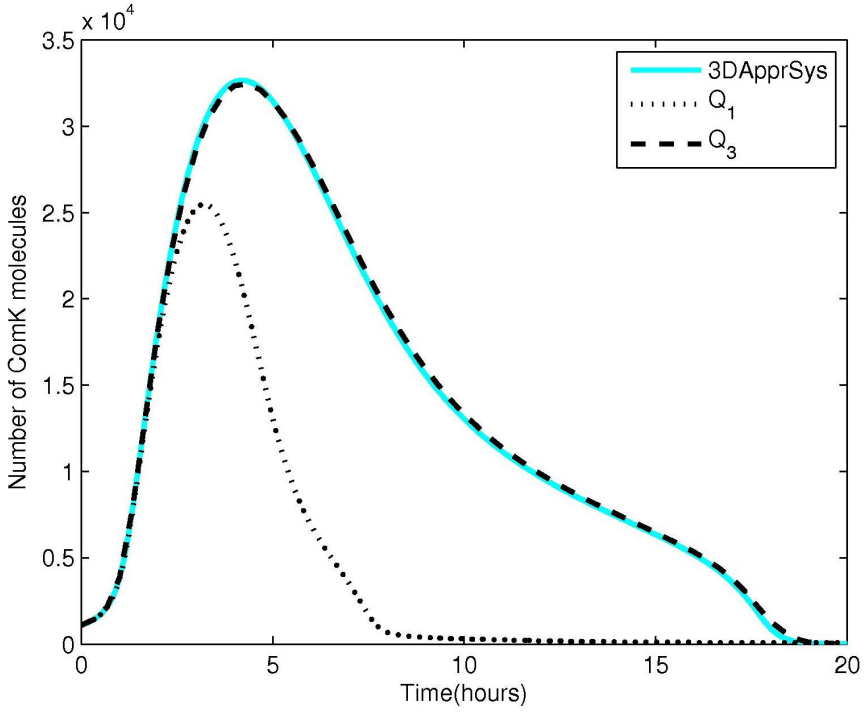


FIGURE 3.11: Competence duration using Q_3 (the 2D approximation) and Q_1 (the adiabatic approximation) in comparison with the 3D deterministic model (3DApprSys).

given in Table 3.1):

$$\begin{aligned} Q_0 &= 0 \\ Q_1 &= F(K, S, 0, 0) \\ Q_2 &= F(K, S, \frac{dQ_1}{dK}, \frac{dQ_1}{dS}) \\ Q_3 &= F(K, S, \frac{dQ_2}{dK}, \frac{dQ_2}{dS}) \\ &\dots \end{aligned} \tag{3.13}$$

Γ_k	25000	Δ_k	0.1	d_s	3.0	ϵ_q	0.09
Γ_s	20	k_0	0.2	Δ_s	0.1	μ	1
a_k	3.5×10^{-4}	c_s	0.033	ϵ_s	2.25		

TABLE 3.1: The parameters used in 3DApprSys.

Approximate Model	3D model	7D Model
Q_1	83.76	88.92
Q_2	8.2	12.93
Q_3	0.15	0.71

TABLE 3.2: Distance between the approximate models and the 3D, 7D deterministic models.

As a result, we now can find a n^{th} approximate function for Q by applying the iterative procedure above. Putting the function Q back into Equation (3.11) then we obtain a 2D deterministic model (2DDeApprSys) of the 3D deterministic model. Numerical experiments show that Q_n converges rapidly and even Q_2 gives a very good approximation. In particular, Fig. 3.10 shows the three different 2D models corresponding to the three first approximate functions of Q (Q_1,Q_2,Q_3) compared with the 3D deterministic model. It turns out that Q_1 is the same expression as that obtained by setting $dQ/dt \approx 0$ in the 2D naive adiabatic approximation. It is clear that the third approximation Q_3 almost perfectly fits the 3D deterministic model. Indeed, we can quantify the difference between the approximate models and the 3D, 7D deterministic models by simply taking $dist = \int_t \left(\log_{10} \left(\frac{K(t)}{K_0(t)} \right)^2 + \log_{10} \left(\frac{S(t)}{S_0(t)} \right)^2 \right) dt$, where $(K_0(t), S_0(t))$ and $(K(t), S(t))$ represent the simulation trajectories generated by the 3D (7D) deterministic model and the approximate models, respectively. These quantities are given in Table 3.2, in which the distance between the trajectories in Q_3 and the 3D (7D) deterministic model is the smallest, we will therefore take Q_3 as the deterministic approximation to the full system,

the differential equation for the 2D approximate model can be described as follows:

$$\begin{aligned}\frac{dK}{dt} &= f_k(K, S, Q_3(K, S)) \\ \frac{dS}{dt} &= f_s(K, S, Q_3(K, S))\end{aligned} \tag{3.14}$$

where

$$\begin{aligned}f_k(K, S, Q_3(K, S)) &= \frac{k_3}{k_7}\left(k_1 + \frac{k_2 K^2}{k_k^2 + K^2}\right) - \frac{k_{12} K Q_3(K, S)}{\Gamma_k + K} - k_8 K \\ f_s(K, S, Q_3(K, S)) &= \frac{k_5 k_6/k_9}{1 + (K/k_s)^5} - k_{10} S - k_{13}\Gamma_k \frac{S Q_3(K, S)}{\Gamma_k + K} \\ &\quad + k_{-13}(M_T - Q_3(K, S))\end{aligned} \tag{3.15}$$

Figure 3.11 shows a comparison of the competence duration between the naive adiabatic approximate model introduced in Süel et al. (2006, 2007) (Q_1) and the iteratively produced model where $Q_3(K, S)$ is used in the K, S evolution equations. Evidently, the competence duration in the 2D approximation (Q_3) is about ten hours which agrees with that in the full system whereas this duration is only roughly four hours in the adiabatic approximation. This significant discrepancy implies that the naive adiabatic model provides a poor approximation of the original system.

We now need to find out whether or not the initial condition impacts on the competence duration in the reduced deterministic model. In fact, the transient before coming to the excitable state is critical to the initialization of probability but it does not significantly impact on the competence duration once the system enters the competent state. In order to verify this claim, we plot different trajectories generated by the 2D deterministic model at different initial conditions such that the system become competent, we then compute the competence duration for each trajectory and compare this quantity with that computed from the stochastic full system. The trajectories from the stochastic model are also sampled and plotted on the same plane. The initial conditions for the integration are chosen from the simulation data generated by the Gillespie algorithm satisfying $1000 \leq K \leq 1585$, $158 \leq S \leq 794$. In our experiment, We take 50 sampled trajectories and normalize them such that they start from the origin of the axis (Figure 3.12). In the full stochastic model, the competence duration is $9.1 \pm \frac{1.0}{\sqrt{50}}$. The competence duration in the deterministic model is $9.4 \pm \frac{0.6}{\sqrt{50}}$. This means the competence duration in the deterministic model is roughly about 10 hours which agrees with that in the stochastic model.

We now compare the 2D deterministic model (Q_3) with the 2D naive adiabatic model (Q_1) in terms of competence duration with different initial conditions. To do so, we apply the same procedure above for the two deterministic models to produce corresponding trajectories, and plotting them on the same axis (Figure 3.13). For the 2D naive adiabatic model, the competence duration computed was $6.3 \pm \frac{3.5}{\sqrt{50}}$ hours compared to

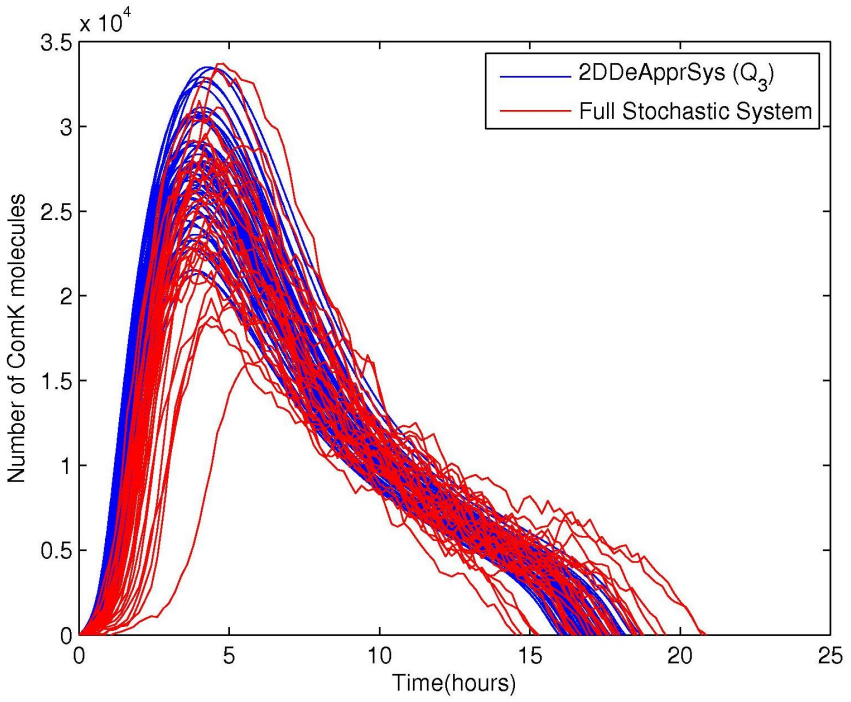


FIGURE 3.12: Competence duration in stochastic and deterministic models.

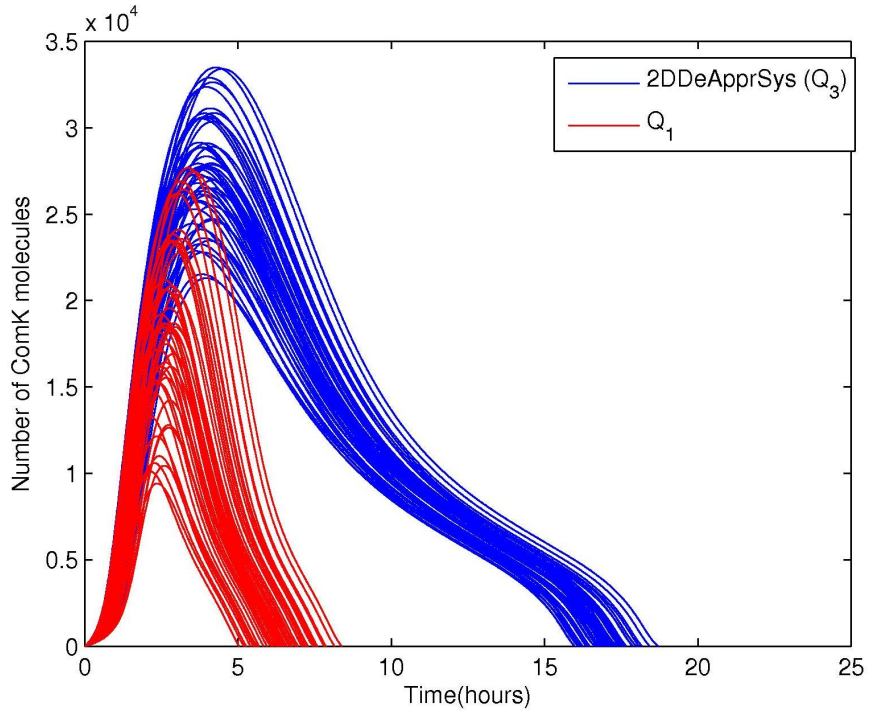


FIGURE 3.13: Competence duration in different deterministic models.

$9.4 \pm \frac{0.6}{\sqrt{50}}$ in the 2D deterministic model. This quantity proves that our reduced model is much better than the 2D naive adiabatic model in terms of competence duration. In other words, the 2D deterministic model captures the dynamics of the cell's behaviour in more accurate way compared to the naive adiabatic model. Moreover, the third ap-

proximation of Q (Q_3) produces a near perfect match in the excitable regime and almost fits the 3D deterministic model all the way back to the vegetative state.

3.3 Summary

In this chapter, I have introduced an effective solution to the model reduction problem in competence regime. I also showed how much improvement we can achieve for the approximate model compared to the adiabatic approximate model. The discrepancy between the full system and the 2D adiabatic approximate model in terms of competence duration in the excitable state shows the weakness in the assumptions of the fast processes. Additionally, it also proves that the model reduction should not be done by an adiabatic approximation alone but need to capture those fast processes in a better way. Our simulation showed that the competence duration computed in this model is smaller than that in the full model by a factor of 3.

Our reduction approach provides a better model in which the competence duration and dynamical behaviour at the excitable regime are conserved. In fact, noise plays a critical role in driving the competence to occur and describing the molecular stochastic processes more realistically. In the following chapter, we are going to find a way of putting the noise back into the reduced system to be able to come up with a stochastic model, and find out if the stochastic model describes the same behaviour as that in the original system.

Chapter 4

Reduction of A 2-Species Model To Track Noise-driven Transition

In the previous chapter, we discussed the model reduction for the deterministic model. The deterministic model, however, does not fully describe systems which are driven by stochastic noise. Stochasticity plays a crucial role in this model as it is the mechanism which drives cells into competence. However, as we will see, modelling the stochastic behaviour is particularly challenging for this model because of the very small number of mRNAs and the existence of a positive feedback loop. Indeed, the standard way of constructing the stochastic model can be performed in two steps: firstly, we replace the chemical master equations by a set of Langevin equations; secondly, we eliminate fast reactions from the reaction rate equations (RRE) to obtain a reduced set of equations describing the dynamics of slowly varying species. However, both these steps were problematic. In fact, for the Langevin equation to approximate the model well requires the molecular number of the species to be significantly large. Unfortunately, the dynamical behaviour relies on a very small number of mRNA, which is then amplified through a positive feedback loop. Thus, eliminating fast variables may result in losing lots of fluctuations which are needed to drive the system to competent state. This consequently leads to the failure of the reduced model to capture the correct noise terms in the original stochastic model and therefore, poorly produces the dynamical behaviour of the system. These issues are briefly described in the following section.

4.1 Discrepancy in Langevin and Gillespie Simulations

As discussed in Chapter 1, the Langevin approximation is used to build up a continuous-time stochastic model of our system. In this section, we will show the fact that the continuous stochastic model may provide a totally different dynamical behaviour in comparison with the Gillespie simulation. In the Langevin simulation (details of the simulation can

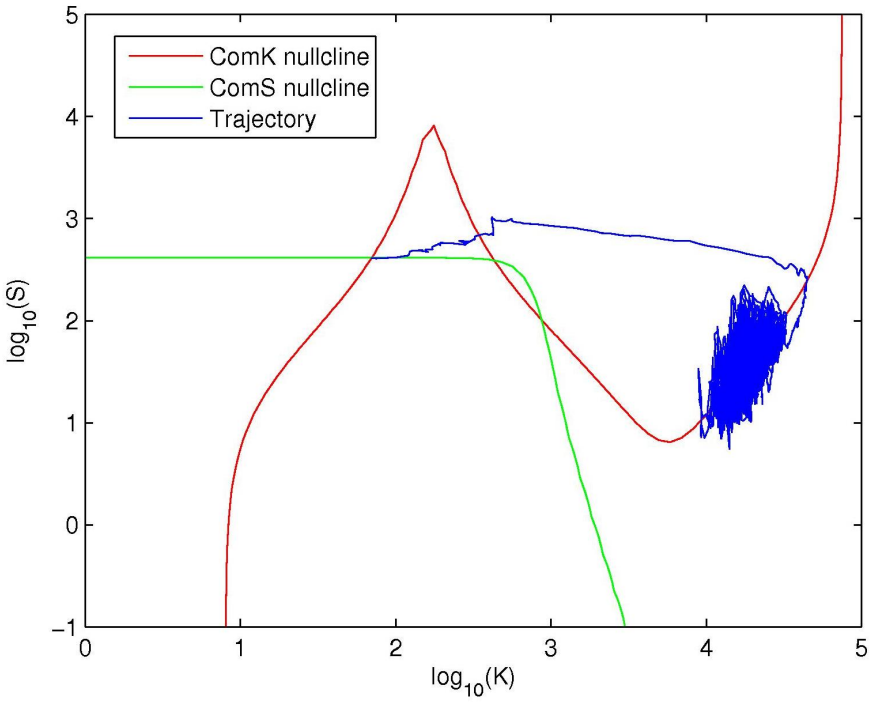


FIGURE 4.1: The trajectories generated by the 7D Langevin simulation.

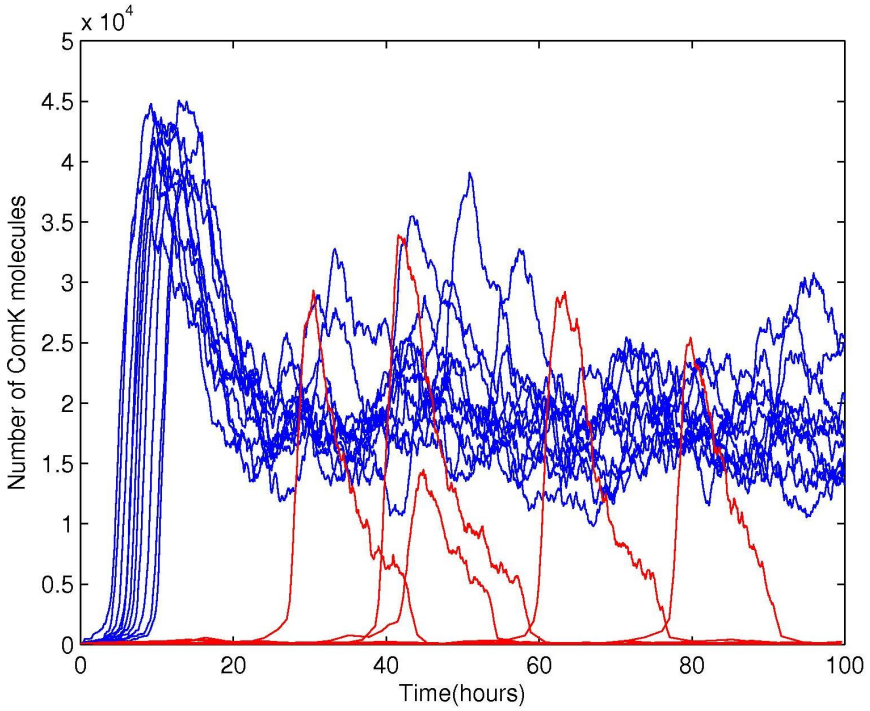


FIGURE 4.2: Competent events in Langevin (blue lines) and Gillespie (red line) simulations.

be found in Appendix A.10), we face the fact that the number of $mRNA_{comS}$ is so small that it may be driven to negative value. However, avoiding this situation may result in incorrect dynamical behaviour. Indeed, the trajectories generated by the stochastic Langevin model are trapped on the way back to the vegetative state (Figure 4.1). In

addition, in this model, cells do not rest in the stationary state but quickly jump into the excitable regime and become competent. In order to statistically estimate how often cells become competent in both models, we run several times simulations for 100 hours with the same initial condition which is chosen at the fixed point of the steady state, then plot the trajectories generated by each model. We found that, in the Langevin model, all the cells enter the competent state while there are only a portion of that population become competent in the Gillespie simulation (Figure 4.2).

In order to avoid this situation, I eliminated the $mRNA_{comS}$ and therefore, reducing the 7D system to a 6D system. However, this reduced model produced a much higher initialization probability compared to the original 7D model. Moreover, the 5D model which is reduced from the 6D model by eliminating both mRNAs does not produce any competent event. This means the fluctuations in $mRNA_{comK}$ which are critical to the switching behaviour has been ignored; therefore, those fluctuations need to be put back into the reduced model. To do this, I apply the same procedure for the reduction but keep the fluctuations in the eliminated variables back into the reduced model. By doing this, the reduced models brought the initialization probability down closer to that in the original model, however, it was still high (all the details of this work are presented in Appendix A.6). This means the noise terms in the stochastic model were still not captured correctly.

In order to address the source of the issue, we will focus our attention on the behaviour of the system near the stable fixed point and the transition beyond the intermediate unstable fixed point to the competent state. To do this, we will be looking at a much simpler noise-driven switching circuit which is extracted from the original system. This is done by just looking at the dynamics of two variables ComK and mRNA while ignoring the effect of the other variables. Even though the behaviour of this bistable model is different from the full model in a long period of time, the dynamics of the system near the fixed point should be similar in both models. Hence, the initialization probability of becoming competent should also be qualitatively the same. As a result, by studying this model, we hope to have isolated the behaviour of the full system we are interested in, and therefore it is easier to address the source of problems which cause the failure of model reduction. In this chapter, as we will see in the Langevin simulation that the very small population of mRNA drives the number of protein negative contributing to the failure of this approximation method. On the other hand, I will show that the adiabatic approximation produces a very poor model even though the decay rates of mRNA are much faster than that of protein. Interestingly, by estimating the size of fluctuation in protein, I have found that the variance of protein is proportional to the square of its molecular number while the Poisson model predicts that it grows linearly to the population of protein. This finding is the key point of incorporating the correct noise into the Langevin equation in order to come up with a better reduced model.

4.2 The Two Species Model

In this section, we are interested in a feedback system in which a protein activates its own transcription as the following chemical reactions:

$$\begin{aligned}
\mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} &\xrightarrow{k_1} \mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} + \mathrm{mRNA}_{\mathrm{comK}} \\
\mathrm{P}_{\mathrm{comK}} &\xrightarrow{f([ComK],k_2,k_k)} \mathrm{P}_{\mathrm{comK}} + \mathrm{mRNA}_{\mathrm{comK}} \\
\mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_3} \mathrm{mRNA}_{\mathrm{comK}} + \mathrm{ComK} \\
\mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_4} \emptyset \\
\mathrm{ComK} &\xrightarrow{k_6} \emptyset
\end{aligned} \tag{4.1}$$

where $f([ComK], k_2, k_k) = \frac{k_2[ComK]^2}{{k_k}^2+[ComK]^2}$.

The first two reactions represent how much $mRNA_{comK}$ is produced from the binding of protein to the promoters on DNA. The next reaction shows how much protein ComK is synthesized from $mRNA_{comK}$. The fourth and fifth reactions represent the linear degradation of the mRNA and protein, respectively. In fact, this model is simplified from the 7D model by setting the variables $MecA$ and $MecA_K$ to their steady values. This is because the numbers don't deviate from the steady state values. Moreover, the system exhibits bistability and transition from a low to a high expression state of ComK which is driven by noise in mRNA levels.

We denote the protein and mRNA as K and m, respectively. As a result, the deterministic differential equations for this model are described as follows:

$$\begin{aligned}
\frac{dK}{dt} &= k_5 + k_3 m - k_6 K \\
\frac{dm}{dt} &= k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2} - k_4 m
\end{aligned} \tag{4.2}$$

The model parameters are given in Table 4.1. The values of parameters k_1, k_2, k_3, k_4 and k_k are the same as that in the original 7D model. The degradation rate in ComK (k_6) was recalculated after setting $MecA$ and $MecA_K$ to their steady values, here I introduce the parameter k_5 ($k_5 = 3.24 \times 10^{-5}$) in order to keep the structure and the position of the fixed points the same as that in the original system. The initial conditions are given in Table 4.2, since P_{comK}^{const} and P_{comK} do not change their concentrations, we set their values to 1 for simplicity. The initial values for $mRNA_{comK}$ and $ComK$ were chosen to be slightly away from the stable fixed point for simulation, the transient was then thrown away. We first plot the nullclines of the system by setting $\frac{dK}{dt} = 0$, $\frac{dm}{dt} = 0$, we

k_1	$0.00021875s^{-1}$
k_2	$0.1875s^{-1}$
k_3	$0.2s^{-1}$
k_4	$0.005s^{-1}$
k_5	$3.2 \times 10^{-5}s^{-1}$
k_6	$1.4704 \times 10^{-4}s^{-1}$
k_k	$5000\,nM$

TABLE 4.1: Model parameters

$[P_{comK}^{const}]$	$1\,nM$
$[P_{comK}]$	$1\,nM$
$[mRNA_{comK}]$	$0\,nM$
$[ComK]$	$20\,nM$

TABLE 4.2: Initial conditions

obtain the following expression of nullclines of K and m, respectively:

$$\begin{aligned} m &= \frac{k_6 K - k_5}{k_3} = h(K) \\ m &= \frac{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}}{k_4} = q(K) \end{aligned} \tag{4.3}$$

In order to find the fixed points of the system, we substitute m from the first equation into the second one in (4.3), this yields

$$k_3 \left(k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2} \right) - k_4 \left(k_6 K - k_5 \right) = 0 \tag{4.4}$$

This cubic equation can be solved in close form. As a result, we obtain three following fixed points:

$$(K, m) = \{(71.0, 0.052), (389.0, 0.285), (54125.0, 39.72)\}$$

The Jacobian matrix for our system has the following form:

$$\mathbf{J} = \begin{array}{c} \\ (K) \\ (m) \end{array} \begin{array}{c} \begin{array}{cc} (K) & (m) \end{array} \\ \begin{pmatrix} \frac{\partial ff}{\partial K} & \frac{\partial ff}{\partial m} \\ \frac{\partial gg}{\partial K} & \frac{\partial gg}{\partial m} \end{pmatrix} \end{array}$$

where $ff = k_5 + k_3 m - k_6 K$, $gg = k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2} - k_4 m$. As a result, we obtain:

$$\mathbf{J} = \begin{array}{c} \\ (K) \\ (m) \end{array} \begin{array}{c} \begin{array}{cc} (K) & (m) \end{array} \\ \begin{pmatrix} -k_6 & k_3 \\ \frac{2k_2 K k_k^2}{(K^2 + k_k^2)^2} & -k_4 \end{pmatrix} \end{array}$$

Hence, the corresponding eigenvalues for the left-most, intermediate and right-most fixed points as shown on the plane are:

$$(e_k, e_s) = \{(-18, -0.3708), (-18.72, 0.36), (-18, -0.504)\}$$

As a result, the left-most and right-most fixed points are classified as stable while the intermediate is unstable fixed points. Consequently, our simple genetic circuit exhibits bistable behaviour with two deterministic steady states (Figure 4.3), a low expression in protein where its molecular number is small ($(K, m) = (71.0, 0.052)$), and a high expression where the protein activates its own transcription generating a switching state at which the protein is present in large numbers ($(K, m) = (54125.0, 39.72)$). Figure 4.4 shows an analysis of regime boundary for the model. In the figure, the regime limited by the blue curve including the low stable fixed point presents a low expression regime, whereas the regime outside of this box containing the other stable fixed point shows a high expression regime. In order to compute the regime boundary, we start with the middle fixed point and compute the Jacobian for that point. Next, we slightly move away from the fixed point in two opposite directions of the eigenvectors and take those as the initial conditions for the function *ode*45 with time step $-\Delta t$. This procedure allow us to generate two "backward" trajectories which form a regime boundary separating the low and high expression level regions.

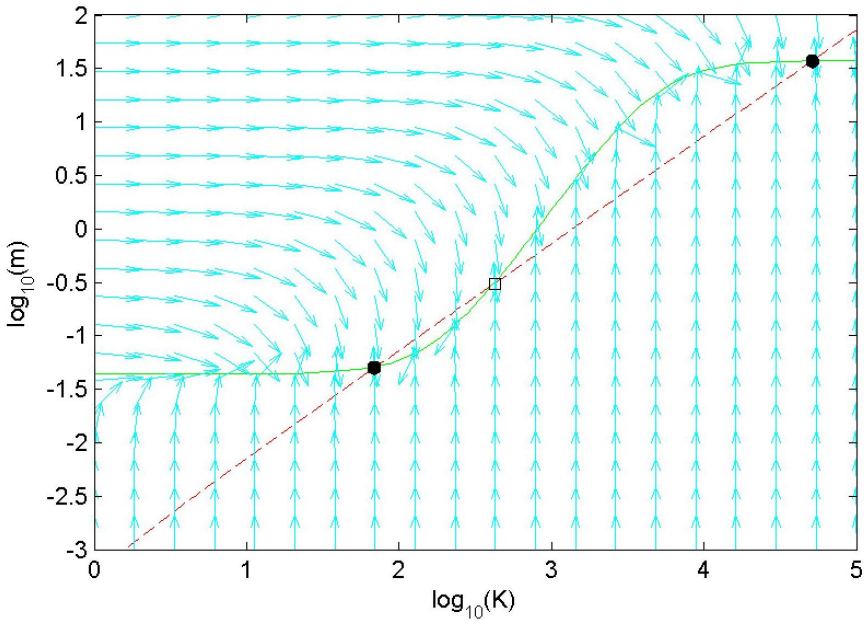


FIGURE 4.3: Nullcline space of the simple model. The dashed line and solid line represent the nullcline of protein and mRNA, respectively. The model exhibits bistable behaviour with two stable fixed points (full black circle) and one intermediate unstable fixed point (empty rectangle). The arrows show the vector field which specifies the direction the trajectories follow.

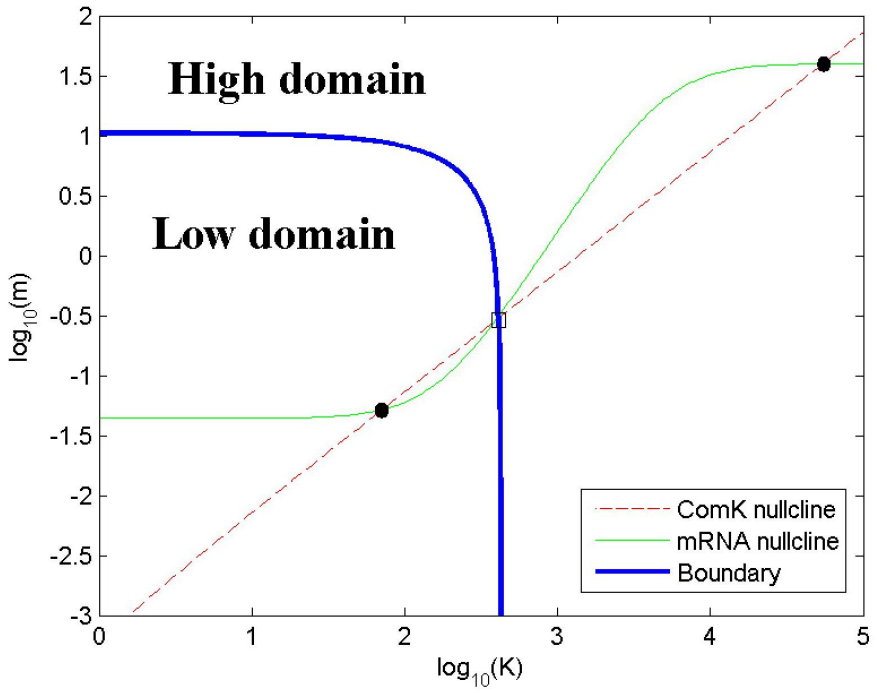


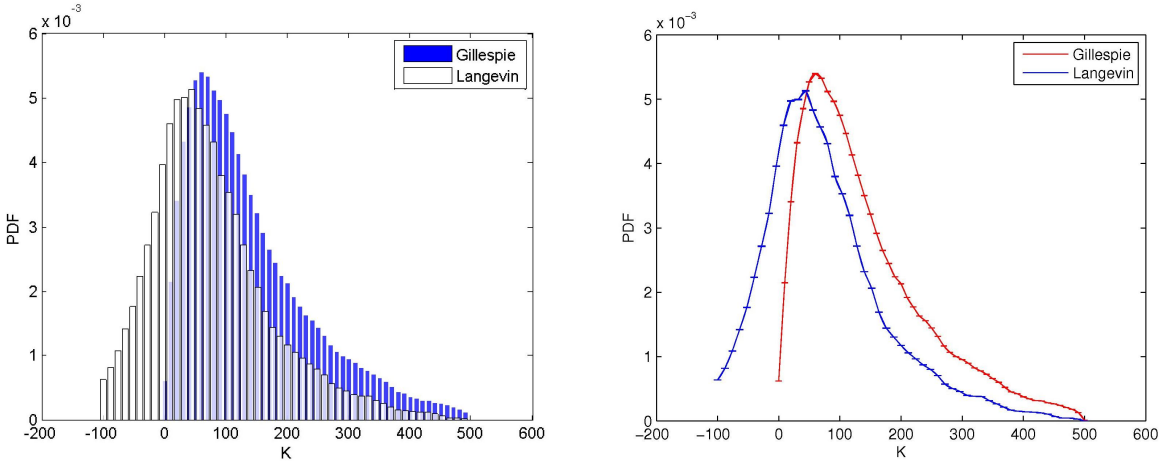
FIGURE 4.4: The boundary between two attraction domains.

4.2.1 Comparing Langevin and Gillespie Simulations

In this context, we are interested in the steady state where the protein exhibits a low expression level. Firstly, we need to verify if the Langevin simulation works well on this model by comparing it to the Gillespie simulation. The 2D Langevin model can be described by the following equations:

$$\begin{aligned} \Delta K &= (k_5 + k_3 m - k_6 K)\,\Delta t + \sqrt{k_3 m + k_5 + k_6 K}\,dW_k \\ \Delta m &= \left(k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2} - k_4 m\right)\Delta t + \sqrt{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2} + k_4 m}\,dW_m \end{aligned} \tag{4.5}$$

where dW_k, dW_m are Wiener processes. Next, we collect all the simulation data and plot PDF fitting histograms for comparison. The PDF fitting histograms are computed as follows: since we are only interested in the low expression region, the simulation data is therefore sampled for this region only. For both Langevin and Gillespie simulations, we use the initial condition given in Table 4.2. We sample M simulations ($M = 100$); for each simulation, the simulation will stop as long as the value of K hits 500 over which ComK reaches high level of expression. In the Gillespie simulation, each sampled data will be put into $N = 50$ bins ranging from 0 to 500 and this process will be iterated for all M simulations. Let s_i be the total number of observations for bin i, $i = 1, 2, \ldots, N$, the PDF for bin i is then estimated as $f_i = \frac{s_i}{\sum_{j=1}^{M} s_j \times w}$, where w is the width of the bin. After this step, we obtain a histogram of the PDF for protein. Similarly, we apply the same procedure for the Langevin simulation, noting the fact that the molecular number



(a) (b)

FIGURE 4.5: Probability density function (PDF) fitting histograms of protein levels in simulations from the Gillespie (4.1) and Langevin simulations (4.5) in the low expression regime (a) and corresponding histogram fitting curves (b).

of protein is driven to negative values which is physically meaningless. However, we still allow this to happen in order to get the distribution of protein. Hence, the bin range for this case is chosen to be from -100 to 500. In order to compare the two histograms, the standard way is to use Kernel density estimation (Rosenblatt, 1956; Bowman, 1984), or smoothing function to approximate the data. In our case, I use function *spline* in Matlab to approximate each histogram by a smooth curve along with error bars (see Figure 4.5)). The figure shows that there is a significant difference between the two curves. In fact, we can quantitatively measure this discrepancy using Kolmogorov-Smirnov test (Schrer and Trenkler, 1995; Justel et al., 1997; Drew et al., 2000). The result shows that the two samples were not drawn from the same distribution (p-value $= 4.74 \times 10^{-6} \ll 0.05$). Consequently, this implies that the dynamical behaviour has not been well approximated by the Langevin approximation.

4.2.2 Tracking Time Scale Separation With Singular Perturbation

In the previous section, I showed that the 2D Langevin model does not produce a good approximate model to the 2D Gillespie model. To solve this problem, we need to see if we can approximate for the deterministic part of the model by looking at the time scale separation of the dynamics. Hence, in this section, we expect to make a better approximation than the adiabatic approximation by applying singular perturbation theory. However, as I will show that this method does not produce a better approximation. Firstly, we speed up the reaction rates k_1, k_2 and k_4 by a factor of $1/\epsilon$ $(0 < \epsilon \leq 1)$; therefore, the evolution of the system can defined as the following slow-fast processes

(Berglund and Gentz, 2003):

$$\begin{aligned} \frac{dK}{dt} &= k_5 + k_3 m - k_6 K \\ \frac{dm}{dt} &= \frac{1}{\epsilon}(h_k - k_4 m) \end{aligned} \tag{4.6}$$

where

$$h_k = k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}, \quad h_k \approx \mathcal{O}(1), \quad k_4 \approx \mathcal{O}(1)$$

We now expand m as a power series in ϵ:

$$m = m^{(0)} + \epsilon m^{(1)} \tag{4.7}$$

In order to find the unknown $m^{(0)}$ and $m^{(1)}$, we substitute the power series (4.7) into the differential equations. This gives us the following expression:

$$\begin{aligned} \frac{\partial m^{(0)}}{\partial t} + \epsilon \frac{\partial m^{(1)}}{\partial t} &= \frac{1}{\epsilon}(h_k - k_4(m^{(0)} + \epsilon m^{(1)})) \\ \frac{dK}{dt} &= k_5 + k_3(m^{(0)} + \epsilon m^{(1)}) - k_6 K \end{aligned} \tag{4.8}$$

We then do the matching for the terms which have the same order of ϵ:

$$\begin{aligned} \mathcal{O}(\epsilon^{-1}) &: h_k - k_4 m^{(0)} = 0 \rightarrow m^{(0)} = \frac{h_k}{k_4} \\ \mathcal{O}(\epsilon^{0}) &: \frac{dm^{(0)}}{dt} = -k_4 m^{(1)} \end{aligned} \tag{4.9}$$

hence,

$$\begin{aligned} \frac{d}{dt}\left(\frac{h_k}{k_4}\right) &= -k_4 m^{(1)} \\ \Rightarrow m^{(1)} &= -\frac{1}{k_4^2}\frac{d(h_k)}{dt} = -\frac{1}{k_4^2}\frac{\partial(h_k)}{\partial K}\frac{dK}{dt} \end{aligned} \tag{4.10}$$

We have

$$\begin{aligned} m = m^{(0)} + \epsilon m^{(1)} &= \frac{h_k}{k_4} - \epsilon \frac{1}{k_4^2}\frac{\partial(h_k)}{\partial K}\frac{dK}{dt} \\ \frac{dK}{dt} &= k_5 + k_3(m^{(0)} + \epsilon m^{(1)}) - k_6 K \\ &= k_5 + k_3 \frac{h_k}{k_4} - \epsilon \frac{k_3}{k_4^2}\frac{\partial(h_k)}{\partial K}\frac{dK}{dt} - k_6 K \end{aligned}$$

Finally,

$$\frac{dK}{dt}\left(1 + \epsilon \frac{k_3}{k_4^2}\frac{\partial(h_k)}{\partial K}\frac{dK}{dt}\right) = k_5 + k_3 \frac{h_k}{k_4} - k_6 K$$

$$\Rightarrow \frac{dK}{dt} = \frac{k_5 + k_3\frac{h_k}{k_4} - k_6 K}{1 + \epsilon\frac{k_3}{k_4^2}\frac{\partial(h_k)}{\partial K}} \tag{4.11}$$

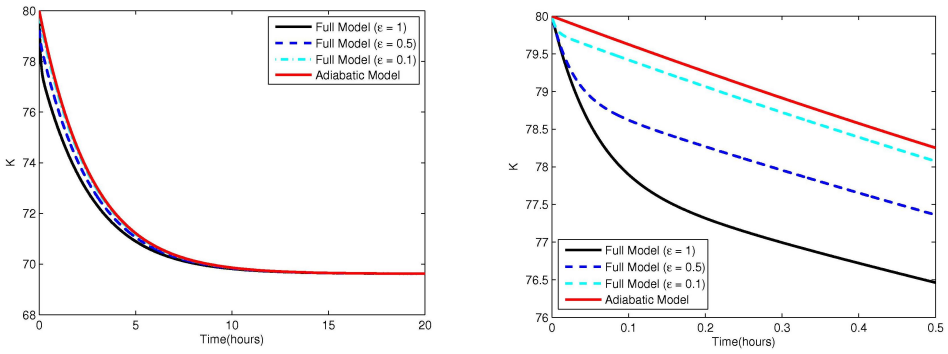


FIGURE 4.6: A comparison between the full model (4.2) and the adiabatic model (4.11) with different values of ϵ. The full model tends to get closer to the adiabatic model as ϵ is small (a); however, the gap between these models in short time scale is still significant (b).

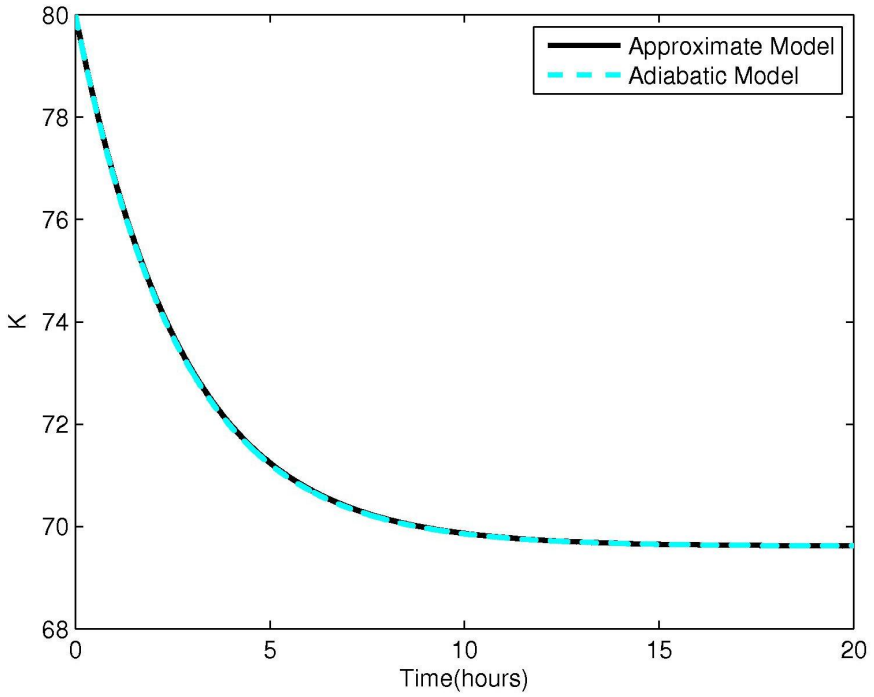


FIGURE 4.7: A comparison between the approximate model (4.11) and the adiabatic model (4.11).

In order to compare the models, we do the simulation for those models using *ode45* function, starting with the initial condition $(K, m) = (80, 0)$ which is set away from the stable fixed point and see how the trajectories is driven back to the fixed point.

Figure 4.6 shows that the full model gets closer to the adiabatic approximate model when ϵ is small; however, there is still a significant difference between these models in short time scale. For this reason, we expect that the approximate model given by (4.11) may provide a better approximation. However, the correction term turns out to be so small that it almost does not make any significant improvement over the adiabatic approximation (Figure 4.7). Indeed, the scaling factor $E = 1 + \epsilon \frac{k_3}{k_4^2} \frac{\partial(h_k)}{\partial K}$ in equation (4.11) does not significantly contribution to the correction term in the approximate model ($E \approx 1$). In fact, we notice that if $\frac{dX}{dt} = f + \eta$ where η is the noise term, then $\eta \propto \sqrt{f}$. On the other hand, our differential equation is scaled by E: $\frac{dX}{dt} = Ef + \sqrt{Ef}\eta$ where $\sqrt{Ef} \propto \sqrt{f}$ since $E \approx 1$, hence the approximation will not work. In fact, the singular perturbation theory works well for the system in long time scale; however, it would never capture the dynamics of system in very short time scale unless we either start the simulation with the initial conditions at which both models are matched or make ϵ extremely small. However, this means the adiabatic model is good enough for the approximation. This result also implies that the time scales are not split out; therefore, the singular perturbation theory does not work properly.

In the next section, I will first compare the model distributions by simulating the dynamics to confirm the time-scales are not separated enough. As a result, we need to find a way of quantifying the noise and incorporating it back into the system. To do so, I demonstrate an empirical method of computing the size of fluctuation near the fixed point and show that the size of fluctuation is proportional to the number of reactants. This analysis is presented in section 4.3.2.

4.3 Stochastic Model Reduction

4.3.1 Time Scale Separation For The Reduction of 2D to 1D Langevin Model

In the previous experiment, the Langevin simulation breaks down due to the very small number of mRNA population having been driven to negative values. Hence, we will try to adiabatically eliminate this small variable and check if this method gives a model that is free from negative protein numbers. We notice that the mRNA lifetimes are shorter than protein lifetimes ($\frac{k_4}{k_6} = 34 \gg 1$), we therefore can assume that the mRNA quickly reaches equilibrium, this yields:

$$m^* = \frac{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}}{k_4} \tag{4.12}$$

$$\frac{dK}{dt} = k_5 + k_3 m^* - k_6 K \tag{4.13}$$

We then get a description of a 1D Langevin equation as follows:

$$dK = \left(k_5 + k_3 \left(\frac{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}}{k_4} \right) - k_6 K \right) dt + \sqrt{k_5 + k_3 \left(\frac{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}}{k_4} \right) + k_6 K}\, dW_k$$
$$m^* = \frac{k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}}{k_4} \tag{4.14}$$

The simulation result shows that there are no competent events in the 1D Langevin

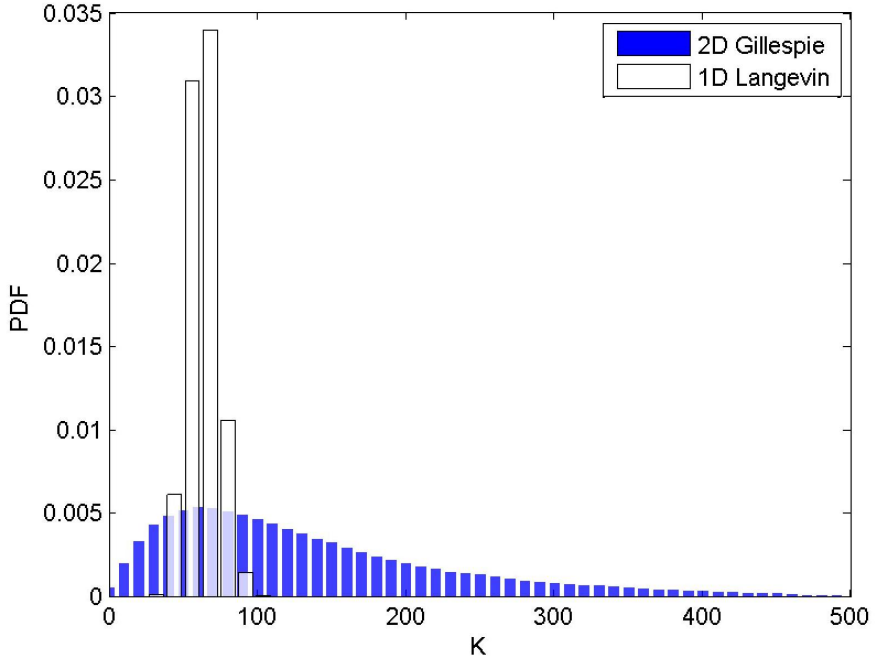


FIGURE 4.8: PDF fitting histograms in the 1D Langevin and 2D Gillespie models.

simulation (Figure 4.8). It means the adiabatic approximation does not capture the noise-driven transitions in this model as shown in the results of the 2D Gillespie model. It also implies that the fluctuation in the mRNA which is generated in the 2D Langevin model significantly contributes to the switching behaviour of the system. Even though the decay rate of mRNA is about 30 times faster than that of the protein, the production rate of mRNA ($k_3 = 0.2$) is 1000 times faster than the decay rate of protein ($k_6 = 1.4704 \times 10^{-4}$); therefore, time scales of mRNA and protein are not completely separated. Consequently, eliminating mRNA results in losing lots of fluctuation which is needed for the transition to occur.

In this section, I have shown that the fluctuation in the 2D Langevin model has not been captured correctly due to the small number of mRNA as well as the impact of the positive feedback mechanism. However, eliminating the mRNA may result in losing the

fluctuation which is needed for driving the system to the high expression regime. For this reason, it is necessary to explore the fluctuations near the steady state in the 2D Gillespie model whereby we would hope to construct the correct noise for the stochastic model. By doing this, I have found that the mean change in mRNA computed from the CME was significantly different from that in the corresponding reaction rate equations. Moreover, the fluctuation computed from the simulation data differs from that obtained from the linear noise approximation method by a factor of 4. This is the key issue which prevented us from producing a good approximate reduced model. However, trying to fix the mean of mRNA could change the characteristics of the fixed points in the original model; therefore, this method is unable to apply in our case (All the details of this work are performed in Appendix A.8). In order to preserve the structure of the fixed points, we still need to do the model reduction using the adiabatic approximation, and try to fit the fluctuation in the stochastic reduced model by using fitting curves. The reason why we are doing this is because we would like to use the results from this study as a guide for the 7D wild-type model. In this analysis, I will show that the standard deviation is proportional to the number of reactants $\sigma_X \propto X$ instead of the usual $\sigma_X \propto \sqrt{X}$ standard deviation in the Langevin equation. By using fitting curves, I concluded that the diffusion coefficient in the stochastic reduced model could be approximated by a quadratic curve, the fluctuation is therefore tunable such that it can provide a better approximation to the original 2D model. In the following section, I will show how to compute the empirical fluctuation from the simulation data.

4.3.2 Fluctuation Estimation

In this section, I will try to estimate the size of fluctuation which is needed to be put back into the 1D Langevin model (4.14), this may allow us to come up with a better approximate model. To obtain this, our calculation will be based on the simulation data of the full Gillespie model. I sample the simulation data as follows:

1. I run M Gillespie simulations using the reaction scheme as mentioned in section 4.1 with the initial condition given by the fixed point. For each run, we stop the simulation as soon as the molecular number of protein exceeds 500. This is the threshold over which the system enters the high expression state.

2. I collect and put the simulation data into N separate bins according to different values of protein K (notice that we are only interested in the values of the protein and mRNA, not the time step). In particular, each bin $i = 1, 2, \ldots N$, contains a particular value of protein K_i and a set of all possible values of $mRNA$ with respect to K_i (we don't count the frequency of $mRNA$). Let L_i be the total number of $mRNA$ values in bin i, then the value of an instance of mRNA j belonging to bin i is denoted as m_{ij} where $j = 1, 2, \ldots L_i$. For each bin $i = 1, 2, \ldots N$, we compute the expected change in protein K_i in time step Δt denoted as ΔK_i that is determined by the propensity functions in

which the protein gets involved. According to this, the expected change in K_i given m_{ij} in a time interval Δt denoted as ΔK_i^j can be estimated as $\Delta K_i^j = (k_5 + k_3 m_{ij} - k_6 K_i)\Delta t$. In our case, I take Δt to be the same as that in the 1D Langevin model (4.14). Since Δt is the same in both Gillespie and 1D Langevin models, the only comparable term would be $k_5 + k_3 m_{ij} - k_6 K_i$. Thus, we can ignore Δt and re-define ΔK_i^j as follows:

$$\Delta K_i^j \equiv k_5 + k_3 m_{ij} - k_6 K_i \tag{4.15}$$

where $j = 1, 2, \ldots L_i$. The variance of ΔK_i is then computed by the following equation:

$$\sigma^2_{\Delta K_i} = E((\Delta K_i - \langle \Delta K \rangle)^2) = \frac{1}{L_i} \sum_{j=1}^{L_i} (\Delta K_i^j - \langle \Delta K_i \rangle)^2 \tag{4.16}$$

Here, $\langle \Delta K_i \rangle = \frac{1}{L_i} \sum_{j=1}^{L_i} \Delta K_i^j$. On the other hand, since CME can be described by a Poisson process; therefore, the variance in protein caused by this process is given as follows:

$$\Sigma_i^2 = k_5 + k_3 \langle m_i \rangle + k_6 K_i \tag{4.17}$$

Here, the mean of mRNA for each bin i is computed as

$$\langle m_i \rangle = \frac{1}{L_i} \sum_{j=1}^{L_i} m_{ij} \tag{4.18}$$

As a result, the size of fluctuation for this particular data bin is given by

$$\Sigma = \sqrt{\Sigma_i^2 + \sigma^2_{\Delta K_i}} \tag{4.19}$$

In our simulation, we start with $M = 100, N = 500$. Figure 4.9 shows the estimated square of size of fluctuation (Σ^2) in protein which we can fit by a polynomial fitting curve. Since we are only interested in fitting the part of the curve which account for the tail of the probability distribution, we will do the fitting for $100 \leq K \leq 500$. Figure 4.9 shows the fitting curves where we fit the empirical curve by a cubic and quadratic curves. Those two fitting curves are defined as follows:

$$y_0 = a_0 K^3 + a_1 K^2 + a_2 K + a_3$$
$$y_1 = b_0 K^2 + b_1 K + b_2$$

where $a_0 = 4.8 \times 10^{-9}$, $a_1 = -3.3 \times 10^{-6}$, $a_2 = 0.001$, $a_3 = 0.057$, $b_0 = 1.1 \times 10^{-6}$, $b_1 = -0.00014$, $b_2 = 0.15$. In fact, we can measure the error for goodness of fit between the fitting curve (f) and the empirical curve (Σ^2) as follows:

$$Err = \sqrt{\sum_{i=100}^{500} ({\Sigma_i}^2 - f_i)^2} \tag{4.20}$$

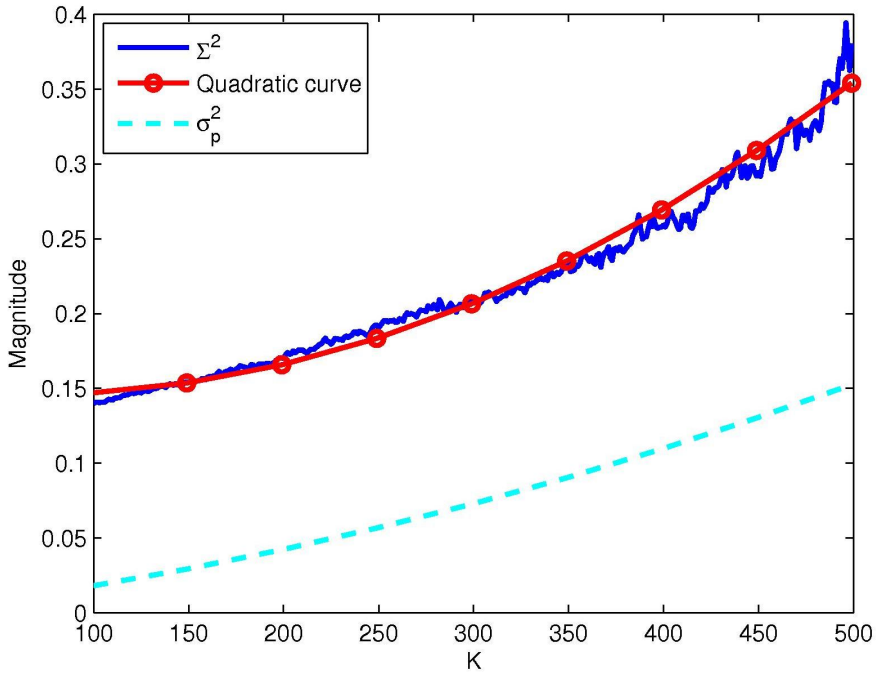


FIGURE 4.9: The square of size of fluctuation given by (4.19) is well fitted by a quadratic curve. This means the noise term in the reduced Langevin model should be proportional to the number of ComK.

My calculation shows that the errors computed for the cubic and quadratic curves are $Err = 0.12$ and $Err = 0.16$, respectively. This means the quadratic fitting curve is only 40% less accurate than the cubic fitting curve; therefore, the empirical curve can be reasonably approximated by a quadratic curve which is simpler than the cubic curve. We also notice that the empirical curve does not grow linearly to K; therefore, it should not be approximated by a linear line.

In this section, I have presented a method of approximating the fluctuation using simulation data. The result has shown that the size of fluctuation can be fit by a quadratic curve. In the next section, I will reduce our model to a 1D model by doing the adiabatic approximation for mRNA. I then approximate the fluctuation by solving the Fokker-Planck equation for this 1D model, and show that the diffusion coefficient in the corresponding Langevin equation is also reasonably fit by a quadratic curve.

4.4 Fluctuation Exploration Using The Fokker-Planck Equation

In this section, we will try to figure out how much noise is needed for the reduced system in order to reproduce the original dynamical behaviour. To do so, we start with the Langevin equation of protein when applying the adiabatic approximation for mRNA.

This equation is described as follows:

$$dK = f(K)dt + \sigma_k\, dW \tag{4.21}$$

where

$$f(K) = k_5 + k_3\left(\frac{k_1 + \frac{k_2 K^n}{k_k{}^n + K^n}}{k_4}\right) - k_6 K \tag{4.22}$$

As a result, the corresponding Fokker-Planck equation is given by:

$$\frac{\partial P(K,t)}{\partial t} = -\frac{\partial}{\partial K}\left(P(K,t)f(K)\right) + \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(P(K,t)D(K)\right) \tag{4.23}$$

Here $P(K,t)$ is the density function and $D(K) = \sigma_k^2$. At the steady state we have $\frac{\partial P(K,t)}{\partial t} = 0$; therefore, equation (4.23) now becomes:

$$-\frac{\partial}{\partial K}\left(P_s^{FP}(K)f(K)\right) + \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(P_s^{FP}(K)D(K)\right) = 0 \tag{4.24}$$

hence,

$$\frac{\partial}{\partial K}\left[-P_s^{FP}(K)f(K) + \frac{1}{2}\frac{\partial}{\partial K}\left(P_s^{FP}(K)D(K)\right)\right] = 0 \tag{4.25}$$

This means the bracketed term is independent of K. Since the probability vanishes for very large and small values of K, the bracketed term has to be zero. As a result, we are left with the first order ordinary differential equation:

$$\frac{\partial}{\partial K}\left(P_s^{FP}(K)D(K)\right) = 2P_s^{FP}(K)f(K) \tag{4.26}$$

or

$$D(K) = \frac{M^{-1}\exp\left(\int_K \frac{2f(K')}{D(K')}\,\mathrm{d}K'\right)}{P_s^{FP}(K)} \tag{4.27}$$

where M is a normalization constant such that $\int_0^{500} P_s^{FP}(K)\,\mathrm{d}K = 1$. Thus, from 4.30 we have

$$M^{-1}\int_0^{500}\frac{\exp\left(\int_0^{500}\frac{2f(K')}{D(K')}\,\mathrm{d}K'\right)}{D(K)}\,\mathrm{d}K = 1 \tag{4.28}$$

thus,

$$M = \int_0^{500}\frac{\exp\left(\int_0^{500}\frac{2f(K')}{D(K')}\,\mathrm{d}K'\right)}{D(K)}\,\mathrm{d}K \tag{4.29}$$

In fact, the probability distribution $P_s^{FP}(K)$ can be obtained from the histogram of the probability distribution of protein that we computed in section 4.2.1; therefore, from (4.30) we can estimate $D(K)$ using the following iterative formulas:

$$D^{(n+1)}(K) = \frac{M^{(n)^{-1}} \exp\left(\int_0^K \frac{2f(K')}{D^{(n)}(K')}\,\mathrm{d}K'\right)}{P_s^{FP}(K)}, \quad n = 0, 1, \ldots \tag{4.30}$$

where $D^{(0)}(K)$ is a guess function (in our case, I chose $D^{(0)}(K) = \Sigma^2$ where Σ is size of fluctuation defined by 4.19, and $M^{(n)}$ is computed as follows,

$$M^{(n)} = \int_0^{500} \frac{\exp\left(\int_0^{500} \frac{2f(K')}{D^{(n)}(K')}\,\mathrm{d}K'\right)}{D^{(n)}(K)}\,\mathrm{d}K \tag{4.31}$$

As a result, the procedure can be described in three steps:

1. Set initial condition $D^{(0)}(K) = \Sigma^2$, $n = 0$.

2. For each $K = 1, 2, \ldots, 500$, we compute

$$M^{(n)} = \int_0^{500} \frac{\exp\left(\int_0^{500} \frac{2f(K')}{D^{(n)}(K')}\,\mathrm{d}K'\right)}{D^{(n)}(K)}\,\mathrm{d}K$$

then update

$$D^{(n+1)}(K) = \frac{M^{(n)^{-1}}}{P_s^{FP}(K)} \exp\left(\int_0^K \frac{2f(K')}{D^{(n)}(K')}\,\mathrm{d}K'\right)$$

In this step, every time we update $D^{(n+1)}(K)$ $(K = 1, 2, \ldots, 499)$, this value will be used to compute $D^{(n+1)}(K+1)$.

3. If $\max_K \left\{ \left| \frac{D^{(n+1)}(K) - D^{(n)}(K)}{D^{(n)}(K)} \right| \right\} < \epsilon$ (I chose $\epsilon = 10^{-4}$) then stop, otherwise set $n = n + 1$ and return to step 2.

Figure 4.10 shows the empirical function $D(K)$ given by (4.30) with initial condition $D^{(0)}(K) = \Sigma^2$, and the corresponding fitting curves which have the following form:

$$z_0 = \alpha_0 K^3 + \alpha_1 K^2 + \alpha_2 K + \alpha_3$$
$$z_1 = \beta_0 K^2 + \beta_1 K + \beta_2$$

where $\alpha_0 = 1.3 \times 10^{-5}$, $\alpha_1 = -0.0094$, $\alpha_2 = 2.2$, $\alpha_3 = -150$, $\beta_0 = 0.0024$, $\beta_1 = -0.99$, $\beta_2 = 110$. The errors for goodness of fit for the cubic and quadratic fitting curves are

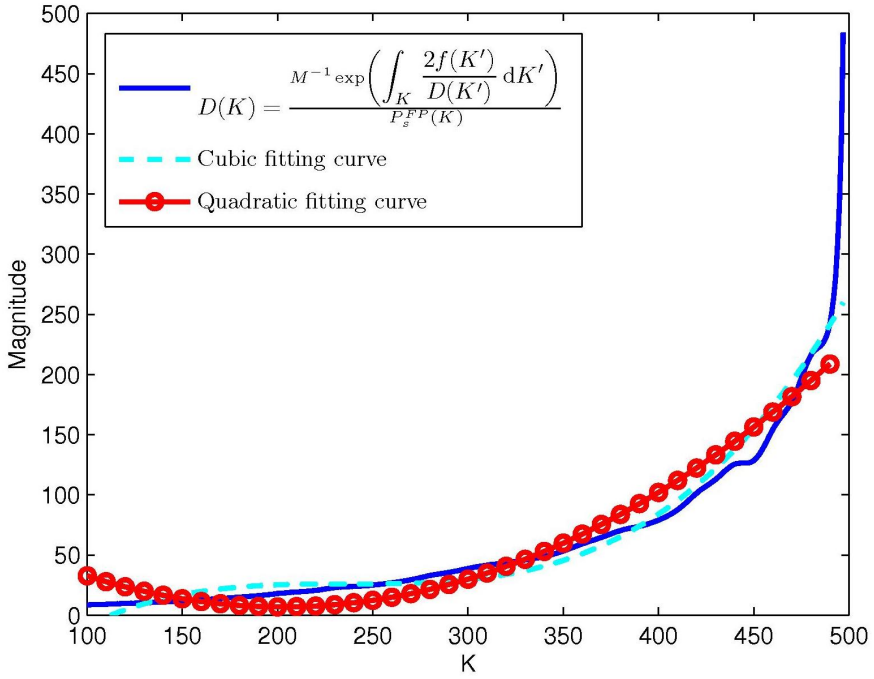


FIGURE 4.10: Fluctuation fitting curve.

$Err = 335.55$ and $Err = 460.2$, respectively. Similarly, the quadratic fitting curve is about 37% less accurate than the cubic fitting curve. For this reason, we can take the quadratic fitting curve as a relatively good approximation for $D(K)$.

In fact, the diffusion coefficient $D(K) = \beta_0 K^2 + \beta_1 K + \beta_2$ with values of the three parameters (β_0,β_1,β_2) given above does not guarantee to produce a best approximate model. Thus, the appropriate solution is to optimize those parameters such that we can obtain a good approximation. This parameter optimization is detailed in the following section.

4.5 Probability Distribution Fitting With Tunable Noise

In this section, I will show how to construct a 1D stochastic reduced model called as the 1D Modified Langevin model. This model is built up by applying the adiabatic approximation for the drift coefficient, and the diffusion coefficient defined by $D(K)$ is supposed to be fit by a quadratic curve. The deterministic description for the reduced model is described as follows:

$$dK = f(K)dt + \sigma_k\, dW \tag{4.32}$$

where

$$f(K) = k_5 + k_3\left(\frac{k_1 + \frac{k_2 K^n}{{k_k}^n + K^n}}{k_4}\right) - k_6 K \tag{4.33}$$

The variance $D(K) = \sigma_k^2$ can be written as follows,

$$D(K) = \beta_0 K^2 + \beta_1 K + \beta_2 \tag{4.34}$$

We now compute the stationary probability distribution at the steady state by solving the following Fokker-Planck equation:

$$\frac{\partial P(K,t)}{\partial t} = -\frac{\partial}{\partial K}\left(P(K,t)f(K)\right) + \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(P(K,t)D(K)\right) \tag{4.35}$$

At the steady state we have $\frac{\partial P(K,t)}{\partial t} = 0$; therefore, equation (4.33) now becomes:

$$-\frac{\partial}{\partial K}\left(P_s^{FP}(K)f(K)\right) + \frac{1}{2}\frac{\partial^2}{\partial K^2}\left(P_s^{FP}(K)D(K)\right) = 0 \tag{4.36}$$

hence,

$$\frac{\partial}{\partial K}\left[-P_s^{FP}(K)f(K) + \frac{1}{2}\frac{\partial}{\partial K}\left(P_s^{FP}(K)D(K)\right)\right] = 0 \tag{4.37}$$

As a result, the solution for (4.37) is

$$P_s^{FP}(K) = \frac{\mathcal{N}^{-1}\exp\left(\int_K \frac{2f(K')}{D(K')}\,\mathrm{d}K'\right)}{D(K)} \tag{4.38}$$

where $\mathcal{N}$ is a normalization constant. From (4.34) and (4.38), we obtain the following expression for $P_s^{FP}(K)$.

$$P_s^{FP}(K) = \frac{\mathcal{N}^{-1}\exp\left(\int_K \frac{2\left(k_5 + \frac{k_3\left(k_1 + \frac{k_2 K'^2}{k_k^2 + K'^2}\right)}{k_4} - k_6 K'\right)}{\beta_0 K'^2 + \beta_1 K' + \beta_2}\,\mathrm{d}K'\right)}{\beta_0 K^2 + \beta_1 K + \beta_2} \tag{4.39}$$

In fact, we are only interested in approximating the tail of the distribution which drives the transition to the high expression regime; therefore, we only need to do the integration for $100 \leq K \leq 500$. For each set of (β_0,β_1,β_2), the solution for $P_s^{FP}(K)$ can be computed using Mathematica, the result is then compared with that obtained from the 2D Gillespie model. We then choose the values of (β_0,β_1,β_2) which best approximates the PDF of the original 2D model. In our case, I choose $0.001 \leq \beta_0 \leq 0.1$, $0 \leq \beta_1 \leq 1$, $1 \leq \beta_2 \leq 100$. My calculation shows that the best values of parameters are:

$$\beta_0 = 0.011, \quad \beta_1 = 0, \quad \beta_2 = 61 \tag{4.40}$$

Figure 4.11 show a comparison between the 1D Modified Langevin and 2D Gillespie models in terms of probability density functions with respect to different sets of parameters

(β_0,β_1,β_2). Since $\beta_1 = 0$, $D(K)$ is reduced to the following form:

$$D(K) = \beta_0 K^2 + \beta_2 \tag{4.41}$$

As we can see in the Figure 4.12, the PDFs near the tail of the distributions seems to be

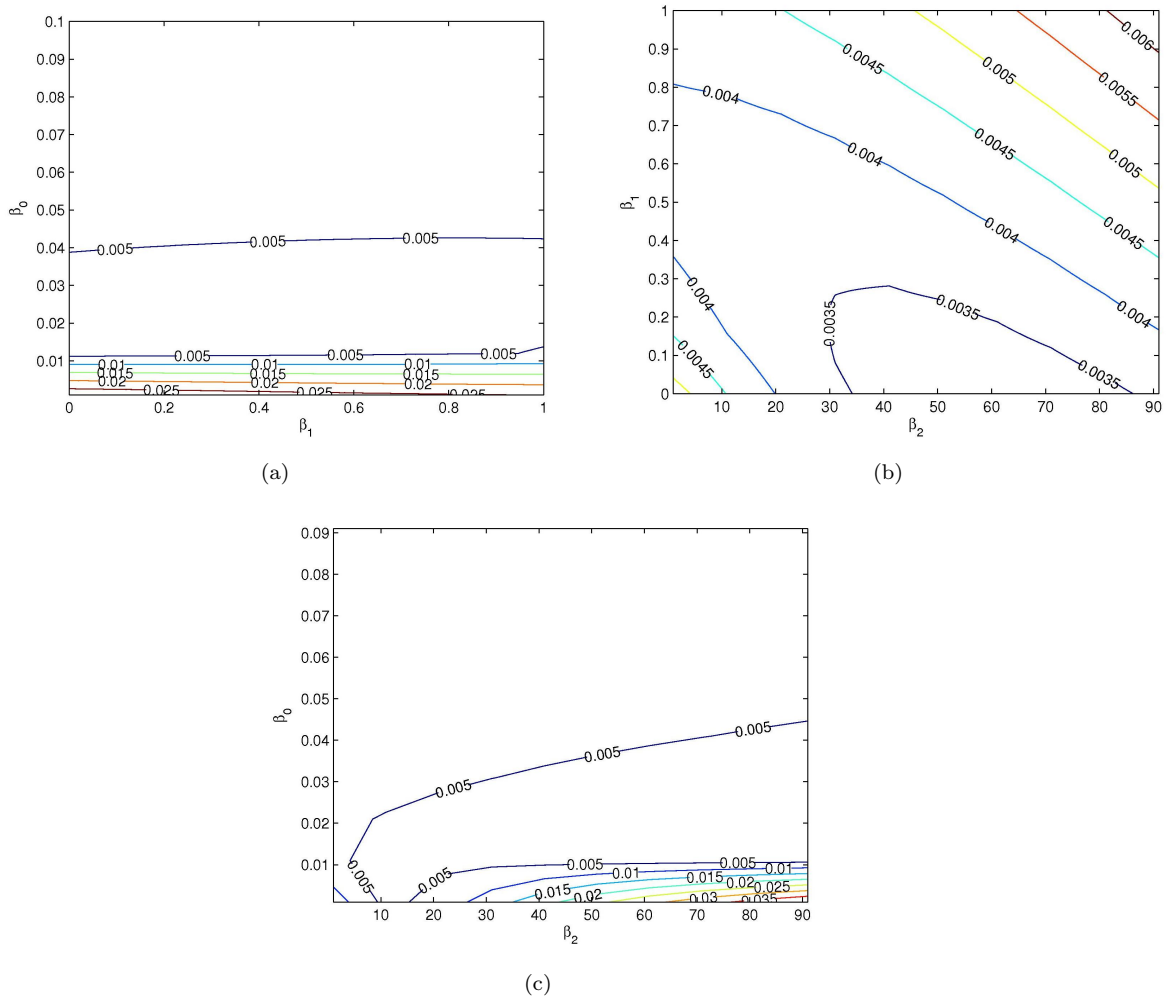


FIGURE 4.11: Contour plots of the distance between two distributions for sets of parameters ($\beta_2 = 61$, β_0, β_1) (a), ($\beta_0 = 0.011$, β_1, β_2) (b) and ($\beta_1 = 0$, β_0, β_2) (c).

similar in both models, but they do not perfectly match due to the fact that the mean of mRNA in the 2D Gillespie model does not match that in the 1D Modified Langevin model as mentioned earlier.

4.6 Summary

In this chapter, I have presented a series of methods to capture the correct noise in the 2-species model which was derived from the full 7D model. By studying this simple model and focusing on the transition beyond the intermediate fixed point to the high expression

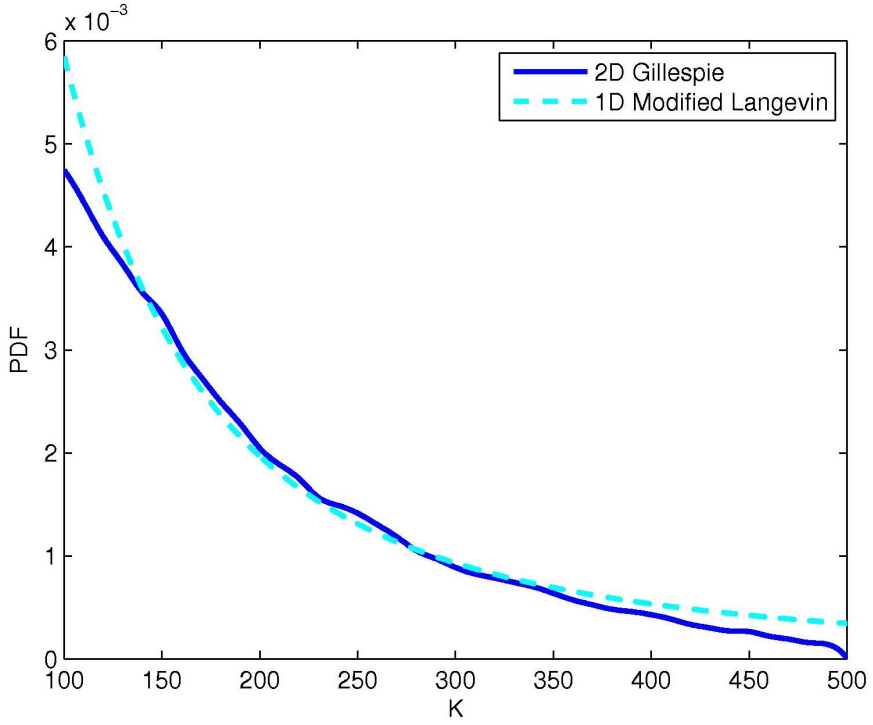


FIGURE 4.12: Probability density functions in the two models with respect to $\beta_0 = 0.011, \quad \beta_1 = 0, \quad \beta_2 = 61$.

regime, I hoped to correctly estimate the fluctuation which is then reflected in the full 7D system. In this model, I have shown that the Langevin method does not produce a good approximation of the full model. The failure of the Langevin approximation comes from different sources. Firstly, the size of fluctuations in the Langevin model drives the number of species negative, which is physically meaningless. Secondly, the system behaviour can not be generated by a Langevin equation with the usual variance describing the fluctuations to be proportional to the number of reactants. In fact, it has been found that the variance is proportional to the square of the number of reactants. More importantly, we have found that the mean of mRNA computed in the simulation deviates significantly from that obtained from the ODE. This is because the process of averaging over the non-linear propensity function was incorrect when reducing from the CME to ODE. Consequently, this prevents us from getting the fluctuation correct for the reduced Langevin model. On the other hand, fixing the mean of mRNA could result in losing the structure of the fixed points in the original model. For this reason, I decided to do the adiabatic approximation for the mRNA in order to keep the fixed points consistent, and try to fit the tail of the probability density function in the stochastic reduced model with tunable noise. In fact, the tunable noise allows us to obtain an approximate model which has the PDF closer to that in the original model. With these results, we can apply the tunable noise for our original 7D wild-type model whereby we hope to produce a better approximate model. In the equation (4.41), we notice that for K large, then $\beta_0 K^2 + \beta_2 \approx \beta_0 K^2$; therefore, we can use this simple form for

the diffusion coefficients of the Langevin equation in the 7D wild-type model. In this case, we temporarily ignore the effect of $MecA$, $MecA_K$ and $MecA_S$ by setting the off-diagonal entries of the covariance matrix for those variables to zero. Consequently, the stochastic reduced model can be obtained by plugging the artificial tunable noise into the model. In the next chapter, we will discuss the Fokker-Planck equation for the wild-type reduced model in order to explore the characteristics of fluctuations which induce the dynamical behaviour of the bacteria.

Chapter 5

A 2D Fokker-Planck Approximation To the Wild-Type Chemical Master Equation

In this chapter, we introduce in the Langevin description, a noise that reflects the ratio of variances of the ComK and ComS distributions at the steady state. We then tune the magnitude of this noise so that the stationary distribution, as computed from the solution of the time-dependent Fokker-Planck equation, gives rise to a bimodal distribution of the ComK-ComS variables that is qualitatively similar to the marginal distribution computed from the Gillespie simulation of the complete wild-type model described in section 2.2. The similarity between the 7D model and the reduced 2D stochastic model is quantified by comparing the corresponding probability density functions (PDFs) computed from each model, using Jensen-Shannon divergence for two distributions.

In the previous chapter, we have introduced the 2D switching model which was derived from the 7D wild-type system. In this model, we have studied the noise driven trajectories that are drawn away from the low expression fixed point by the dynamics, and found that the noise term in the reduced stochastic model was proportional to the number of protein. In this chapter, we hope to apply this finding to our 2D reduced model in order to reproduce the dynamical behaviour observed in the 7D wild-type model.

5.1 The Fokker-Planck Equation For 2D Model

In our 2D reduced wild-type system (Chapter 3, equation (3.14)), we have a 2D Langevin equation for $\mathbf{X} = [K\ S]^\intercal$ which can generally be described as follows:

$$d\mathbf{X} = \mathbf{f}dt + \boldsymbol{\mu}d\mathbf{W} \tag{5.1}$$

where $\mathbf{f} = [f_k(K,S,t)\ f_s(K,S,t)]^\intercal$, $\boldsymbol{\mu} = \begin{pmatrix} \mu_k(K,S,t) & 0 \\ 0 & \mu_s(K,S,t) \end{pmatrix}$

As we mentioned above, we have found in the 2D switching model that, when reducing from a 2D Langevin model to a 1D Langevin model, the noise term was proportional to the number of protein. For this reason, we expect that we can use this as an assumption for the 2D model described above; therefore, we have $\mu_k(K,S,t) = \sigma_k K, \quad \mu_s(K,S,t) = \sigma_s S$. For simplicity, we set the off-diagonal terms of the covariance matrix to zero; therefore, the cross-derivatives terms in the Fokker-Planck equation can be removed. As a result, the full form of the Fokker-Planck equation is given by:

$$\begin{aligned}\frac{\partial P(K,S,t)}{\partial t} = &-\left[\frac{\partial}{\partial K} f_k(K,S,t)P(K,S,t) + \frac{\partial}{\partial S} f_s(K,S,t)P(K,S,t)\right] \\ &+ \frac{1}{2}\left[\frac{\partial^2}{\partial K^2}\mu_k^2(K,S,t)P(K,S,t) + \frac{\partial^2}{\partial S^2}\mu_s^2(K,S,t)P(K,S,t)\right] \end{aligned} \tag{5.2}$$

We now convert K,S to 10-base logarithm phase by setting $x = \log_{10}(K), \quad y = \log_{10}(S)$. Hence:

$$\begin{aligned}\frac{\partial \widetilde{P}(x,y,t)}{\partial t} = &-\left[\frac{\partial}{\partial x}\frac{\widetilde{f_k}(x,y,t)\widetilde{P}(x,y,t)}{10^x \ln(10)} + \frac{\partial}{\partial y}\frac{\widetilde{f_s}(x,y,t)\widetilde{P}(x,y,t)}{10^y \ln(10)}\right] \\ &+ \frac{1}{2}\left[\frac{\partial^2}{\partial x^2}\frac{\sigma_k^2\widetilde{P}(x,y,t)}{\ln^2(10)} + \frac{\partial^2}{\partial y^2}\frac{\sigma_s^2\widetilde{P}(x,y,t)}{\ln^2(10)}\right] \end{aligned} \tag{5.3}$$

where $\widetilde{f_{k,s}}(x,y,t) = f_{k,s}(10^x, 10^y, t), \quad \widetilde{P}(x,y,t) = P(10^x, 10^y, t)$. We now set $a(x,y,t) = \frac{\widetilde{f_k}(x,y,t)}{10^x \ln(10)}$, $b(x,y,t) = \frac{\widetilde{f_s}(x,y,t)}{10^y \ln(10)}$, $\alpha(x,y,t) = \frac{\sigma_k^2}{\ln^2(10)}$, $\beta(x,y,t) = \frac{\sigma_s^2}{\ln^2(10)}$, Equation (5.3) can be rewritten as below:

$$\begin{aligned}\frac{\partial \widetilde{P}(x,y,t)}{\partial t} = &-[\frac{\partial}{\partial x}a(x,y,t)\widetilde{P}(x,y,t) + \frac{\partial}{\partial y}b(x,y,t)\widetilde{P}(x,y,t)] \\ &+ \frac{1}{2}[\frac{\partial^2}{\partial x^2}\alpha(x,y,t)\widetilde{P}(x,y,t) + \frac{\partial^2}{\partial y^2}\beta(x,y,t)\widetilde{P}(x,y,t)] \end{aligned} \tag{5.4}$$

In order to numerically integrate this equation, we use *finite difference method* (Press et al., 1992) in which a function $f(x,y,t)$ is represented by its values at the discrete sets of points:

$$\begin{aligned} x_j &= x_0 + j\Delta_x \qquad j = 0, 1, \ldots, J \\ y_l &= y_0 + l\Delta_y \qquad l = 0, 1, \ldots, L \\ t_n &= t_0 + n\Delta_t \qquad n = 0, 1, \ldots, N \end{aligned}$$

Δ_x and Δ_y are grid spacings along x-axis and y-axis, respectively; Δ_t is time step. From now on, we will write $a_{j,l}^n$ for $a(x_j, y_l, t_n)$, $b_{j,l}^n$ for $b(x_j, y_l, t_n)$, $\alpha_{j,l}^n$ for $\alpha(x_j, y_l, t_n)$, $\beta_{j,l}^n$ for $\beta(x_j, y_l, t_n)$ and $\widetilde{P}_{j,l}^n$ for $\widetilde{P}(x_j, y_l, t_n)$. Since then, the differential terms given by (5.4)

can be estimated as follows:

$$\begin{aligned}
\frac{\partial}{\partial x}a(x,y,t_n)\widetilde{P}(x,y,t_n) &= \frac{a^n_{j+1,l}\widetilde{P}^n_{j+1,l} - a^n_{j-1,l}\widetilde{P}^n_{j-1,l}}{2\Delta_x} \\
\frac{\partial}{\partial y}b(x,y,t_n)\widetilde{P}(x,y,t_n) &= \frac{b^n_{j,l+1}\widetilde{P}^n_{j,l+1} - b^n_{j,l-1}\widetilde{P}^n_{j,l-1}}{2\Delta_y} \\
\frac{\partial^2}{\partial x^2}\alpha(x,y,t_n)\widetilde{P}(x,y,t_n) &= \frac{\alpha^n_{j+1,l}\widetilde{P}^n_{j+1,l} - 2\alpha^n_{j,l}\widetilde{P}^n_{j,l} + \alpha^n_{j-1,l}\widetilde{P}^n_{j-1,l}}{\Delta_x^2} \\
\frac{\partial^2}{\partial y^2}\beta(x,y,t_n)\widetilde{P}(x,y,t_n) &= \frac{\beta^n_{j,l+1}\widetilde{P}^n_{j,l+1} - 2\beta^n_{j,l}\widetilde{P}^n_{j,l} + \beta^n_{j,l-1}\widetilde{P}^n_{j,l-1}}{\Delta_y^2} \\
\frac{\partial\widetilde{P}(x,y,t_n)}{\partial t} &= \frac{\widetilde{P}^{n+1}_{j,l} - \widetilde{P}^n_{j,l}}{\Delta_t}
\end{aligned}$$

Equation (5.4) now becomes:

$$\begin{aligned}
\frac{\widetilde{P}^{n+1}_{j,l} - \widetilde{P}^n_{j,l}}{\Delta_t} = &\left(-\frac{b^n_{j,l+1}}{2\Delta_y} + \frac{\beta^n_{j,l+1}}{2\Delta_y^2}\right)\widetilde{P}^n_{j,l+1} + \left(\frac{b^n_{j,l-1}}{2\Delta_y} + \frac{\beta^n_{j,l-1}}{2\Delta_y^2}\right)\widetilde{P}^n_{j,l-1} + \left(\frac{a^n_{j-1,l}}{2\Delta_x} + \frac{\alpha^n_{j-1,l}}{2\Delta_x^2}\right)\widetilde{P}^n_{j-1,l} \\
&+ \left(-\frac{a^n_{j+1,l}}{2\Delta_x} + \frac{\alpha^n_{j+1,l}}{2\Delta_x^2}\right)\widetilde{P}^n_{j+1,l} - \left(\frac{\alpha^n_{j,l}}{\Delta_x^2} + \frac{\beta^n_{j,l}}{\Delta_y^2}\right)\widetilde{P}^n_{j,l}
\end{aligned}$$

The equation above can be written as $\frac{\widetilde{\mathbf{P}}^{n+1} - \widetilde{\mathbf{P}}^n}{\Delta_t} = \mathbf{A}\widetilde{\mathbf{P}}^n$, in which $\widetilde{\mathbf{P}} \in \mathbb{R}^{J\times L}$, $\mathbf{A} \in \mathbb{R}^{JL\times JL}$ is a constant, very sparse matrix. As $n \to \infty$ then $\widetilde{\mathbf{P}}^{n+1} - \widetilde{\mathbf{P}}^n = 0 \to \mathbf{A}\widetilde{\mathbf{P}}^n = 0$; therefore, the solution is to find the eigenvector of matrix $\mathbf{A}$ corresponding to its eigenvalue being zero. In our case, we use a grid including 100×100 blocks ($J = L = 100$) for doing the discretization. All the derivatives are then computed on a 10-base logarithm phase plane (details of the method used for approximation can be found in the Appendix A.9). In the following section, we will derive the probability density function by solving the Fokker-Planck equation for the 2D approximate model.

5.2 Probability Density Function in The 2D Approximate Model

As mentioned in Chapter 3, the 2D deterministic approximate model (2DDeApprSys) (3.14) can be described as the following differential equations:

$$\begin{aligned}
\frac{dK}{dt} &= f_k(K,S,Q_3(K,S)) \\
\frac{dS}{dt} &= f_s(K,S,Q_3(K,S))
\end{aligned} \tag{5.5}$$

We now analyze the stability of the model by looking at the fixed points. Figure 5.1 shows the location of the nullclines in the 10-base logarithm K-S phase plane, together

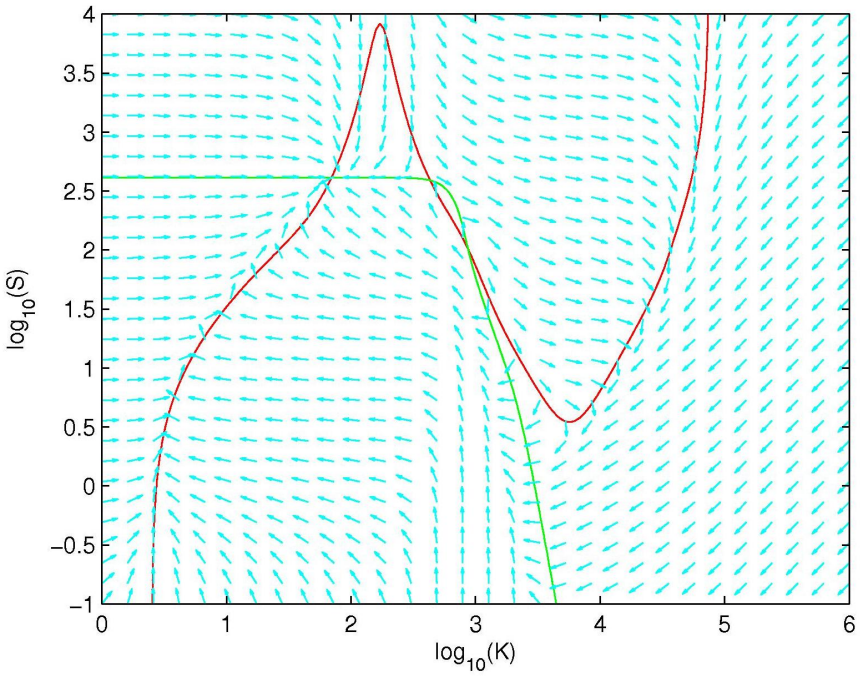


FIGURE 5.1: Phase-plane analysis of the 2DDeApprSys.

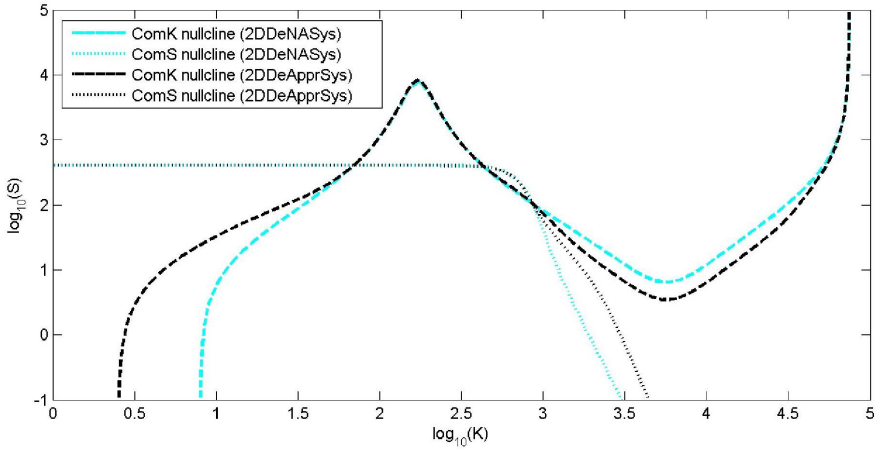


FIGURE 5.2: The nullclines in the 2D naive adiabatic model (2DDeNASys) and the 2D approximate model (2DDeApprSys).

with the vector field presenting the directions of trajectories which follow. There are three fixed points which are the intersections of the nullclines, the left-most of which is stable and corresponds to the vegetative fixed point. The two others are unstable (the middle one is a saddle and the right-most one is an unstable fixed point). Even though the nullclines in this model are shifted with respect to the 2D naive adiabatic model (2DDeNASys), the characteristics and positions of those fixed points are the same in both models (Figure 5.2); therefore, cells also spend most of time residing in the region which is near the stable fixed point. Consequently, it would be nice if we can capture the probability distribution of the time cells spend in different regimes by solving the Fokker-Planck equation. As I mentioned earlier, since we don't know how to calculate

the correct size of the noise, we are forced to tune it empirically. Let us consider a corresponding stochastic process driven by noise of the approximate model below:

$$\begin{aligned} dK &= f_k(K, S, Q_3(K, S))dt + \mu_k dw_k \\ dS &= f_s(K, S, Q_3(K, S))dt + \mu_s dw_s \end{aligned} \tag{5.6}$$

Where dw_k, dw_s are Wiener processes, and we set $\mu_k = \sigma_k K$, $\mu_s = \sigma_s S$. This is known as a stochastic version of the 2D deterministic model (2DDeApprSys), we name this model as 2DStoApprSys (see the model structure 3.5). The magnitudes σ_k, σ_s of the noise terms are chosen to obtain the stationary probability distribution qualitatively similar to that of the CME. Additionally, the initialization probability of competent events computed from the stochastic model should also quantitatively be preserved. However, this probability is very sensitive to the switching behaviour driven by the noise terms. In other words, a slight change in coefficients σ_k, σ_s leads to a significant change in the initialization probability. This is because of the exponential sensitivity of the tail of the probability distribution to the noise-driven switching state in our particular circuit (Mehta et al., 2008). Consequently, I have tried the simulation with different values of σ_k, σ_s and quantify the similarity between the full model and the stochastic model. To do so, I compare the PDFs computed from full discrete model and the stochastic model using Jensen-Shannon divergence (Fuglede and Topsoe, 2004) which is a smoothed version of the Kullback-Leibler divergence (Johnson and Sinanovic, 2001). The PDF of the stochastic model can be obtained by solving the corresponding Fokker-Planck equation while the PDF of the full discrete model is computed from the binned data. Indeed, I first collect the simulation data obtained from Dizzy (Ramsey et al., 2005) for ComK and ComS, I then create a data grid size 100×100 on a log-scale plane where $0 \leq \log_{10}(K) \leq 5$, $-2 \leq \log_{10}(S) \leq 3.5$. The collected data will be stored in 100×100 bins, each bin is located at $(\log_{10}(K(i,j)), \log_{10}(S(i,j)))$ on the grid ($i = 1, 2, \ldots 100$, $j = 1, 2, \ldots 100$). A particular data point $(\log_{10}(K'), \log_{10}(S'))$ is classified to bin (i, j) if $\log_{10}(K(i,j)) - \Delta_K/2 \leq \log_{10}(K') \leq \log_{10}(K(i,j)) + \Delta_K/2$, and $\log_{10}(S(i,j)) - \Delta_S/2 \leq \log_{10}(S') \leq \log_{10}(S(i,j)) + \Delta_S/2$, where Δ_K and Δ_S are grid spacings along axis K and S, respectively. The PDF for point $(\log_{10}(K(i,j)), \log_{10}(S(i,j)))$ is computed by dividing the number of data points classified to that bin by the total of data points, then dividing by the area of the bin (which is $\Delta_K \times \Delta_S$). After this step, we obtain a PDF for all points on the grid and therefore, is comparable to that computed from the Fokker-Planck equation in the stochastic model. Since the PDFs are stored as 2D matrices, I therefore convert them to 1D vectors so that we can use Jensen-Shannon divergence (from now on, we will use only one index to describe 1D probabilities). In order to apply this method, we need to normalize the PDFs to obtain the corresponding probabilities P and Q where $\sum_i P(i) = 1$, $\sum_i Q(i) = 1$. Suppose that P and Q are the probabilities in the full discrete model and the Fokker-Planck equation, respectively; the Jensen-Shannon distance between P and Q is then quantified as $D(P,Q) = \frac{D(P||M)+D(Q||M)}{2}$, where $M = \frac{P+Q}{2}$, and $D(P||M)$, $D(Q||M)$ are Kullback-Leibler divergences. However,

there are two issues we have to deal with: firstly, since P is computed from the binned data, there are many points on the grid at which $P = 0$ (unseen events), this makes it difficult to compare distributions that predict non-zero probability for unseen events; secondly, there are many points on the grid at which values of Q are extremely small (ranging from 10^{-20} to 10^{-30} in our simulation). However, we should never predict the derived probability of an event that is completely impossible; therefore, we set them all to zero to avoid unnecessary bias when comparing with unseen events in P. As a result, this again makes the divergence infinite. To solve those issues, we must take into account the possibility of unseen events by doing a pre-processing procedure for P and Q as follows (we take P for example):

1. Let $S = \{i | 1 \leq i \leq 10000\}$, $U = \{i | P(i) < \epsilon\}$ (in our case, I choose $\epsilon = 10^{-20}$), and $V = S \setminus U$.

2. We define $P' : S \mapsto [0, 1]$ such that, $P'(i) = \epsilon$ for $i \in U$, otherwise $P'(i) = P(i) - \epsilon \frac{|U|}{|V|}$.

By doing this, we don't need to re-normalize P' since $\sum_i P'(i) = \sum_i P(i) + \epsilon\,|U| - \epsilon \frac{|U|}{|V|}\,|V| = \sum_i P(i) = 1$, and similar for Q'.

For each pair of parameters (σ_k, σ_s), we compute $D(P, Q)$ and sort them in descending order, we then choose the pair of parameters corresponding to the smallest distance. The pair of parameters chosen for our experiment satisfies $0.005 \leq \sigma_k \leq 0.02$ and $0.001 \leq \sigma_s \leq 0.02$. Figure 5.3 shows the distance between the PDFs in the full model and stochastic model as a function of the noise terms.

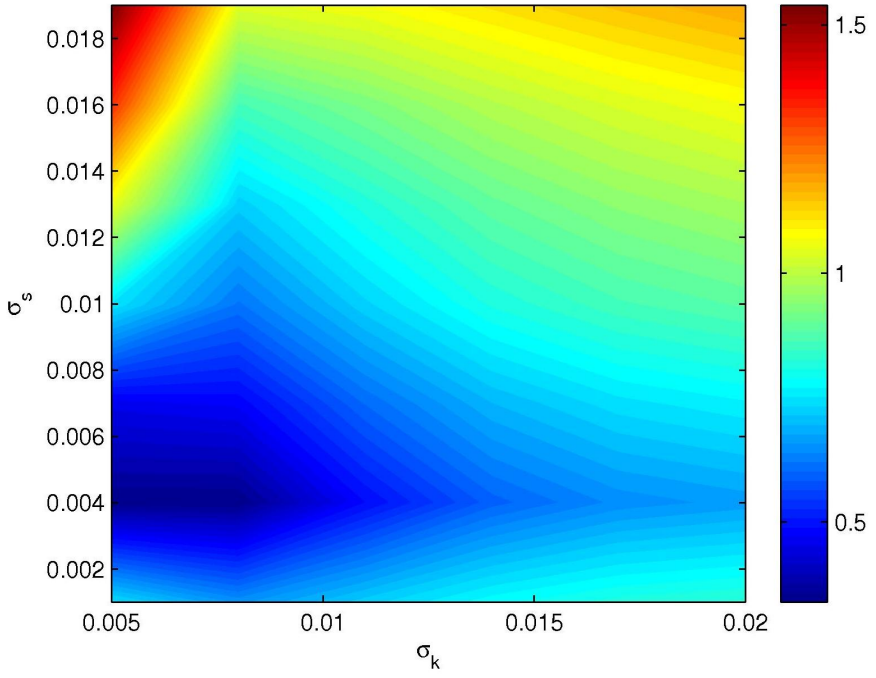
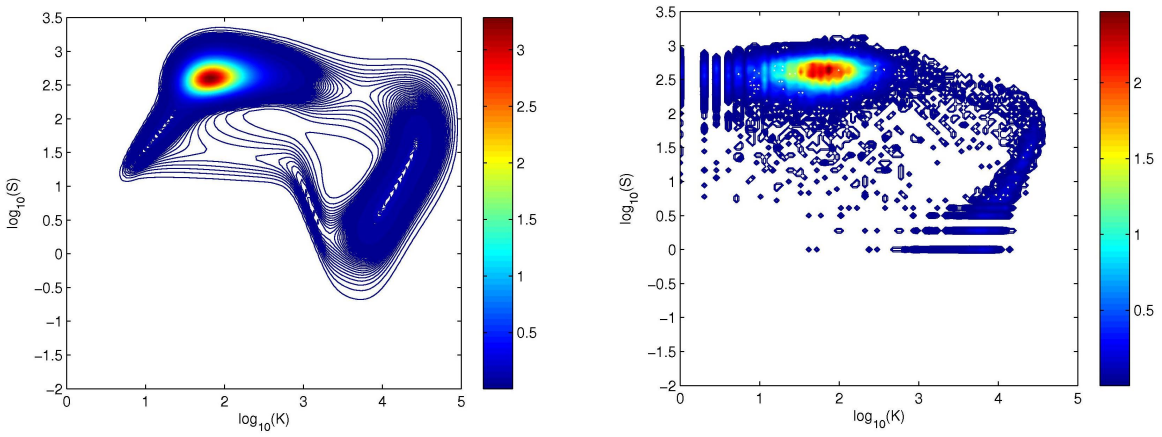


FIGURE 5.3: Contour plot of distance as a function of σ_k and σ_s.

In our experiment, I found that the stochastic model provides the best approximation with $\sigma_k = 0.008$, $\sigma_s = 0.005$. As evidence, I show in Figure 5.4 2D contour plots of the probability density function by solving the Fokker-Planck equation for the 2D stochastic model (2DStoApprSys) described by (5.6), which are similar to those generated from the CME by the Gillespie algorithm. In particular, both models produce similar bimodal distributions (the dense areas) that are characteristics of the cell counts in the vegetative and competent states obtained in Süel et al. (2006, 2007). Moreover, the initialization probability was computed at $0.0043 \pm 4.2 \times 10^{-4}$ which is just slightly smaller than that in the full system ($P_{init} = 0.0076 \pm 2.3 \times 10^{-4}$).



(a) (b)

FIGURE 5.4: Contour plots of probability density function of the 2DStoApprSys (a), and probability distribution generated from the full discrete model (b).

5.3 Summary

To sum up, the objective of this chapter is to show whether or not we can use the tunable noise which was observed in the 2D switching model for our 2D reduced model, in order to obtain a similar stationary probability distribution compared to that in the 7D wild-type system. To do so, we have shown the numerical solution of the Fokker-Planck which captures a bimodal probability distribution of the 2D approximate model, in which it approximates the distribution of species under stochastic evolution. By tuning the noise, we could estimate the similarity between the PDFs obtained from the 2D stochastic and 7D discrete models. Based on the estimated similarity with different values of the tunable noise, we have chosen the noise parameters which best approximates the original 7D model. Moreover, our experiment showed that the approximation has been much improved; particularly, the stationary probability distribution probability is similar to that in the original 7D system, and the probability of initialization is closer to that computed from the 7D system.

The limitation of this work is that we still have yet to show the explicit description of the correct noise terms for the stochastic approximate model. The tunable noise is only derived from first principles by taking into account the dynamical characteristics of the variables being marginalized over. To what extent, the tunable noise induced system may not quantitatively give us the right answer for the dynamics of system, but it has been a significant improvement in the model reduction problem. On the other hand, since the initialization probability of competence is sensitive to the noise terms that induce switching behaviour, it is much more challenging to tune the noise in order to get the right probability distribution. In spite of that, we have shown the tunable noise can be experimentally chosen such that the corresponding stochastic model better approximates the dynamics of the system.

In the next chapter, we are going to study another genetic circuit which is believed to have a similar behaviour to the native wild-type circuit, but provides a less variation in competence durations. This is to evaluate the generalisability of our method to genetic circuits that generate excitable dynamics. However, as we will see that this circuit also faces the same challenges which have been found in the wild-type model.

Chapter 6

The SynExSlow Genetic Circuit

Süel et al. (2007) used the phase diagram described in Chapter 2 to drive the wild-type bacterium into dynamical regimes that are not normally observed in nature to support the model of competence they proposed. Instead of altering the parameters of a given circuit, it is possible to insert a completely new circuit into the cell and design cell behaviour. A theoretical analysis of an alternative circuit showed it to be capable of generating excitable behaviour just like the wild-type circuit. This motivates Cagatay et al. to investigate how such an alternative circuit topology could behave in a cell by engineering the regulatory networks SynEx and SynExSlow. In this chapter, we perform a detailed analysis of this novel circuit, paying particular attention to the issue of model reduction as before. We show that the ODE description from which the dynamical behaviour was derived in Cagatay et al. (2009a) does not naturally follow from a description in term of chemical reactions. We then construct a reaction scheme that allows us to perform stochastic simulations in this model. We also find dynamical behaviour that have not been reported in Cagatay et al. (2009a). The approach of this chapter reverses the presentational order of model description. Here we start with the RRE and infer a CME that can reproduce the RRE that describes the mean of the variables.

6.1 Stochastic Description of SynExSlow

The SynEx circuit as designed was shown to have competence dynamics similar to native cells but displayed a significantly smaller variation in competence durations. However, the competent events were shorter than that in the native circuit. To make this closer to that in native cells, they created the SynExSlow strain derived from the SynEx with longer competent events by competitively interfering with the degradation of ComK by MecA during the competence (Figure 6.1). This delayed exit from competence and therefore made the competence durations longer.

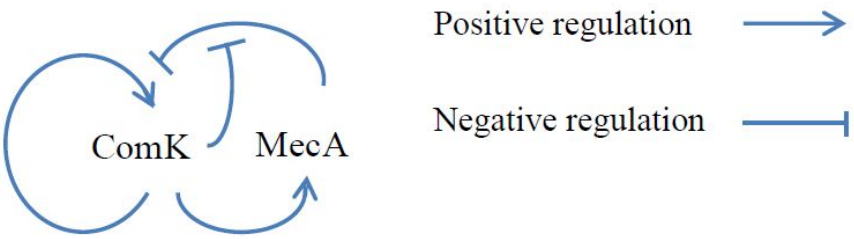


FIGURE 6.1: Topology of SynExSlow strain.

α_k	$0.00875\ molec/s$	β_k	$7.5\ molec/s$	k_k	$5000\ molec$
α_m	$0.075\ molec/s$	β_m	$2.5\ molec/s$	k_m	$2500\ molec$
α_s	$0.5\ molec/s$	β_s	$0.5\ molec/s$	k_s	$500\ molec$
δ_k, δ_s	$2 \times 10^{-6}\ molec^{-1}s^{-1}$	Γ_k	$25000\ molec$	n	2
λ_k, λ_m, λ_s	$10^{-4}\ s^{-1}$	Γ_s	$20\ molec$	p	2

TABLE 6.1: Parameters used in the deterministic equations of the SynExSlow model (Source from Cagatay et al. (2009b)).

6.1.1 Postulating A CME to match The RRE

The deterministic description of the SynExSlow is expressed as follows:

$$\begin{aligned}
\frac{dM}{dt} &= \alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p} - \lambda_m M \\
\frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_k K \\
\frac{dS}{dt} &= \alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s S M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_s S
\end{aligned} \tag{6.1}$$

where M is the total concentration of MecA, while K and S are denoted $[ComK]$ and $[ComS]$, respectively. The model parameters are given in Table 6.1.

In previous chapters, we started with the chemical reactions to derive an ODE description for the system. However, in this section, we will do it backward: we start with the ODE description that the authors provided and try to construct the corresponding stochastic description which describes the model in terms of chemical reactions. Based on the stochastic description of the SynEx provided in Cagatay et al. (2009b). We come

k_1	$0.00022\ s^{-1}$	k_6	$0.2\ s^{-1}$	k_9	$0.2\ s^{-1}$	k_{12}	$0.005\ s^{-1}$
k_2	$0.19\ s^{-1}$	k_7	$0.005\ s^{-1}$	k_{10}	$0.005\ s^{-1}$	k_{13}	$0.0001\ s^{-1}$
k_3	$0.2\ s^{-1}$	k_8	$0.0625\ s^{-1}$	k_{11}	$0.005\ s^{-1}$		

TABLE 6.2: The reaction rates used in the stochastic SynExSlow model.

up with the following chemical reactions (the values of reaction rates taken from the paper are detailed in Table 6.2):

$$
\begin{aligned}
\mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} &\xrightarrow{k_1} \mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} + \mathrm{mRNA}_{\mathrm{comK}} \\
\mathrm{P}_{\mathrm{comK}} &\xrightarrow{f([ComK],k_2,k_k,n)} \mathrm{P}_{\mathrm{comK}} + \mathrm{mRNA}_{\mathrm{comK}} \\
\mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_3} \mathrm{mRNA}_{\mathrm{comK}} + \mathrm{ComK} \\
\mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} &\xrightarrow{k_4} \mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} + \mathrm{mRNA}_{\mathrm{comS}} \\
\mathrm{P}_{\mathrm{comS}} &\xrightarrow{g([ComK],k_5,k_s,n)} \mathrm{P}_{\mathrm{comS}} + \mathrm{mRNA}_{\mathrm{comS}} \\
\mathrm{mRNA}_{\mathrm{comS}} &\xrightarrow{k_6} \mathrm{mRNA}_{\mathrm{comS}} + \mathrm{ComS} \\
\mathrm{P}_{\mathrm{mecA}}^{\mathrm{const}} &\xrightarrow{k_7} \mathrm{P}_{\mathrm{mecA}}^{\mathrm{const}} + \mathrm{mRNA}_{\mathrm{mecA}} \\
\mathrm{P}_{\mathrm{mecA}} &\xrightarrow{h([ComK],k_8,k_m,p)} \mathrm{P}_{\mathrm{mecA}} + \mathrm{mRNA}_{\mathrm{mecA}} \\
\mathrm{mRNA}_{\mathrm{mecA}} &\xrightarrow{k_9} \mathrm{mRNA}_{\mathrm{mecA}} + \mathrm{MecA} \\
\mathrm{mRNA}_{\mathrm{comK}} &\xrightarrow{k_{10}} \emptyset \\
\mathrm{ComK} &\xrightarrow{k_{13}} \emptyset \\
\mathrm{mRNA}_{\mathrm{comS}} &\xrightarrow{k_{11}} \emptyset \\
\mathrm{ComS} &\xrightarrow{k_{14}} \emptyset \\
\mathrm{mRNA}_{\mathrm{mecA}} &\xrightarrow{k_{12}} \emptyset \\
\mathrm{MecA} &\xrightarrow{k_{15}} \emptyset \\
\mathrm{MecA}_{\mathrm{K}} &\xrightarrow{k_{15}} \emptyset \\
\mathrm{MecA}_{\mathrm{S}} &\xrightarrow{k_{15}} \emptyset \\
\mathrm{MecA} + \mathrm{ComK} &\xrightarrow{k_{16}/\Omega} \mathrm{MecA}_{\mathrm{K}} \\
\mathrm{MecA}_{\mathrm{K}} &\xrightarrow{k_{-16}} \mathrm{MecA} + \mathrm{ComK} \\
\mathrm{MecA}_{\mathrm{K}} &\xrightarrow{k_{17}} \mathrm{MecA} \\
\mathrm{MecA} + \mathrm{ComS} &\xrightarrow{k_{18}/\Omega} \mathrm{MecA}_{\mathrm{S}} \\
\mathrm{MecA}_{\mathrm{S}} &\xrightarrow{k_{-18}} \mathrm{MecA} + \mathrm{ComS} \\
\mathrm{MecA}_{\mathrm{S}} &\xrightarrow{k_{19}} \mathrm{MecA}
\end{aligned}
\tag{6.2}
$$

Here, the Hill equations are given by:

$$
\begin{aligned}
f\left([ComK], k_2, k_k, n\right) &= \frac{k_2 [ComK]^n}{{k_k}^n + [ComK]^n} \\
g\left([ComK], k_5, k_s, n\right) &= \frac{k_5 [ComK]^n}{{k_s}^n + [ComK]^n} \\
h\left([ComK], k_8, k_m, p\right) &= \frac{k_8 [ComK]^p}{{k_m}^p + [ComK]^p}
\end{aligned}
$$

In this set of chemical reactions, I assume that $MecA_K$ and $MecA_S$ are degraded at the same rate as that in the degradation of $MecA$, so that we can obtain the deterministic description of the SynExSlow as described in equation (6.1). Indeed, using the same assumption as done with the wild-type circuit, the rates of change in molecular numbers of $MecA_K$ and $MecA_S$ can be described by the following differential equations:

$$\frac{d[MecA_K]}{dt} = k_{16}[ComK][MecA] - k_{-16}[MecA_K] \tag{6.3}$$

$$- k_{17}[MecA_K] - k_{15}[MecA_K] \tag{6.4}$$

$$\frac{d[MecA_S]}{dt} = k_{18}[ComS][MecA] - k_{-18}[MecA_S] \tag{6.5}$$

$$- k_{19}[MecA_S] - k_{15}[MecA_S] \tag{6.6}$$

We now assume that the binding and unbinding processes of proteins and protease complex are very fast so that the rest of the system only responds to the steady-state values of $MecA_K$ and $MecA_S$. Consequently, we set:

$$\frac{d[MecA_K]}{dt} \approx 0 \quad , \quad \frac{d[MecA_S]}{dt} \approx 0$$

therefore,

$$\begin{aligned} [MecA_K] &= \frac{k_{16}[ComK][MecA]}{k_{17} + k_{-16} + k_{15}} = \frac{[MecA][ComK]}{\Gamma_k} \\ [MecA_S] &= \frac{k_{18}[ComS][MecA]}{k_{19} + k_{-18} + k_{15}} = \frac{[MecA][ComS]}{\Gamma_s} \end{aligned} \tag{6.7}$$

We denote M as the total concentration of $MecA$, then we have:

$$[MecA] + [MecA_K] + [MecA_S] = M$$

therefore,

$$[MecA] + \frac{[ComK][MecA]}{\Gamma_k} + \frac{[ComS][MecA]}{\Gamma_s} = M$$

We obtain:

$$[MecA] = \frac{M}{1 + \frac{[ComK]}{\Gamma_k} + \frac{[ComS]}{\Gamma_s}} \tag{6.8}$$

On the other hand, we have the corresponding differential equations for $mRNA_{mecA}$ and $MecA$ as follows:

$$\begin{aligned} \frac{d[mRNA_{mecA}]}{dt} &= k_7[P_{mecA}^{const}] + [P_{mecA}]\frac{k_8[ComK]^p}{{k_m}^p + [ComK]^p} \\ &\quad - k_{12}[mRNA_{mecA}] \\ \frac{d[MecA]}{dt} &= k_9[mRNA_{mecA}] - k_{15}[MecA] \\ &\quad - k_{16}[MecA][ComK] + k_{-16}[MecA_K] \end{aligned}$$

$$+ k_{17}[MecA_K] - k_{18}[MecA][ComS] + k_{-18}[MecA_S] + k_{19}[MecA_S] \tag{6.9}$$

Since the concentrations of promoters do not change through the reactions, we set their concentration to 1 for simplicity. We observe that the mRNA dynamics are faster than that of proteins and tend to reach steady-state values faster than proteins. As a result, we can approximate the effect of mRNA on the protein dynamics by an adiabatic approximation, yielding:

$$[mRNA_{mecA}] = \frac{k_7}{k_{12}} + \frac{k_8[ComK]^p}{k_{12}({k_m}^p + [ComK]^p)} \tag{6.10}$$

Since (6.7) and (6.9) we obtain:

$$\begin{aligned}\frac{d[MecA]}{dt} &= k_9[mRNA_{mecA}] - k_{15}[MecA] - k_{15}[MecA_K] \\ &\quad - k_{15}[MecA_S] \\ &= k_9[mRNA_{mecA}] - k_{15}([MecA] + [MecA_K] + [MecA_S]) \\ &= k_9[mRNA_{mecA}] - k_{15}M \\ &= \frac{k_9k_7}{k_{12}} + \frac{k_9k_8}{k_{12}}\frac{[ComK]^p}{{k_m}^p + [ComK]^p} - k_{15}M\end{aligned}$$

We now set

$$\alpha_m = \frac{k_9k_7}{k_{12}}, \quad \beta_m = \frac{k_9k_8}{k_{12}}, \quad \lambda_m = k_{15} \tag{6.11}$$

therefore,

$$\frac{dM}{dt} = \alpha_m + \frac{\beta_m[ComK]^p}{{k_m}^p + [ComK]^p} - \lambda_m M \tag{6.12}$$

On the other hand, we also have the following differential equations for ComK and ComS:

$$\frac{d[ComK]}{dt} = k_3[mRNA_{comK}] - k_{13}[ComK] - k_{16}[MecA][ComK] + k_{-16}[MecA_K] \tag{6.13}$$

$$\frac{d[ComS]}{dt} = k_6[mRNA_{comS}] - k_{14}[ComS] - k_{18}[MecA][ComS] + k_{-18}[MecA_S] \tag{6.14}$$

The differential equations for $mRNA_{comK}$ and $mRNA_{comS}$ are:

$$\frac{d[mRNA_{comK}]}{dt} = k_1 + \frac{k_2[ComK]^n}{{k_k}^n + [ComK]^n} - k_{10}[mRNA_{comK}] \tag{6.15}$$

$$\frac{d[mRNA_{comS}]}{dt} = k_4 + \frac{k_5[ComK]^n}{{k_s}^n + [ComK]^n} - k_{11}[mRNA_{comS}] \tag{6.16}$$

Since the dynamics of mRNAs are faster than that of proteins, we set $\frac{d[mRNA_{comK}]}{dt} = 0$

and $\frac{d[mRNA_{comS}]}{dt} = 0$, we obtain:

$$[mRNA_{comK}] = \frac{k_1}{k_{10}} + \frac{k_2[ComK]^n}{k_{10}({k_k}^n + [ComK]^n)} \tag{6.17}$$

$$[mRNA_{comS}] = \frac{k_4}{k_{11}} + \frac{k_5[ComK]^n}{k_{11}({k_s}^n + [ComK]^n)} \tag{6.18}$$

From equations (6.17), (6.18), (6.13), (6.14), (6.8) and (6.7), we obtain the following expression for ComK and ComS:

$$\frac{d[ComK]}{dt} = \frac{k_1k_3}{k_{10}} + \frac{k_2k_3/k_{10}[ComK]^n}{{k_k}^n + [ComK]^n} - \frac{k_{16}(k_{17}+k_{15})M[ComK]}{(k_{17}+k_{-16}+k_{15})\left(1+\frac{[ComK]}{\Gamma_k}+\frac{[ComS]}{\Gamma_s}\right)} - k_{13}[ComK]$$

$$\frac{d[ComS]}{dt} = \frac{k_4k_6}{k_{11}} + \frac{k_5k_6/k_{11}[ComK]^n}{{k_s}^n + [ComK]^n} - \frac{k_{18}(k_{19}+k_{15})M[ComK]}{(k_{19}+k_{-18}+k_{15})\left(1+\frac{[ComK]}{\Gamma_k}+\frac{[ComS]}{\Gamma_s}\right)} - k_{14}[ComK]$$

By redefining the variables (K for $[ComK]$ and S for $[ComS]$), we obtain the following differential equations:

$$\begin{aligned} \frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k MK}{1+\frac{K}{\Gamma_k}+\frac{S}{\Gamma_s}} - \lambda_k K \\ \frac{dS}{dt} &= \alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s MS}{1+\frac{K}{\Gamma_k}+\frac{S}{\Gamma_s}} - \lambda_s S \end{aligned} \tag{6.19}$$

where:

$$\begin{aligned} \alpha_k &= \frac{k_3k_1}{k_{10}},\ \beta_k = \frac{k_2k_3}{k_{10}},\ \delta_k = \frac{k_{16}(k_{17}+k_{15})}{k_{17}+k_{-16}+k_{15}},\ \lambda_k = k_{13} \\ \alpha_s &= \frac{k_4k_6}{k_{11}},\ \beta_s = \frac{k_5k_6}{k_{11}},\ \delta_s = \frac{k_{18}(k_{19}+k_{15})}{k_{19}+k_{-18}+k_{15}},\ \lambda_s = k_{14} \end{aligned} \tag{6.20}$$

From (6.12) and (6.19), we obtain the deterministic description of the SynExSlow model described in (6.1). This means the stochastic description above seems to be equivalent to the deterministic description of the SynExSlow. However, in the following section, we will find that the deterministic description can not be obtained from the system of chemical reactions described above.

6.1.2 Inconsistency of Parameter Values of CME

From equations (6.7), (6.11), (6.20) and , we have:

$$\begin{aligned} k_{14} &= \lambda_s = 0.0001 \\ k_{15} &= \lambda_m = 0.0001 \\ k_4 &= \frac{\alpha_s k_{11}}{k_6} = 0.0125 \\ k_5 &= \frac{\beta_s k_{11}}{k_6} = 0.0125 \\ k_{19} + k_{15} &= \delta_s \Gamma_s = 4 \times 10^{-5} \\ k_{17} + k_{15} &= \delta_k \Gamma_k = 0.05 \end{aligned} \tag{6.21}$$

As a result, we obtain $k_{19} = \delta_s \Gamma_s - k_{15} = 4 \times 10^{-5} - 0.0001 = -6 \times 10^{-5} < 0$. In fact, we still can do the Gillespie simulation even in this case where the reaction rate is negative by replacing the reaction by its reversed reaction. However, this leads to an incorrect expression of the propensity function. Thus, the SynExSlow model fails to be reconstructed from the stochastic chemical kinetics. In other words, the SynExSlow model can not be used to describe the molecular dynamics for such physical events which happen inside cells. On the other hand, if the model were correct then the deterministic differential equations obtained above should be approximated by a continuous Markov process that satisfies the following *Chemical Langevin Equation (CLE)* (Gillespie, 2002, 2007; Cazzaniga et al., 2006):

$$\begin{aligned} dM &= \left(\alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p} - \lambda_m M \right) dt + \eta_m dW_m \\ dK &= \left(\alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_k K \right) dt + \eta_k dW_k \\ dS &= \left(\alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s S M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_s S \right) dt + \eta_s dW_s \end{aligned} \tag{6.22}$$

where dW_m dW_k dW_s are standard *Wiener* processes, and:

$$\begin{aligned} \eta_m &= \sqrt{\alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p} + \lambda_m M} \\ \eta_k &= \sqrt{\alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} + \frac{\delta_k K M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} + \lambda_k K} \\ \eta_s &= \sqrt{\alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} + \frac{\delta_s S M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} + \lambda_s S} \end{aligned}$$

In order to understand this model, we need to analyse the stability of the continuous model around the fixed points. Since the model is a 3D model, we therefore try to reduce it to a 2D model using an adiabatic approximation. However, as I will show that this

approximation does not produce a good model.

6.2 The Continuous Model

We re-write the continuous description of the SynExSlow model as follows:

$$\begin{aligned}
\frac{dM}{dt} &= \alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p} - \lambda_m M \\
\frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_k K \\
\frac{dS}{dt} &= \alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s S M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_s S
\end{aligned} \tag{6.23}$$

Since we want to look at the dynamics of the system on (K,S) plane, we will try to eliminate the variable M using an adiabatic approximation. Assuming that the dynamics of M is very fast; therefore its time-dependent evolution can be set to its steady state value ($\frac{dM}{dt} \approx 0$), we obtain:

$$M = \frac{\alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p}}{\lambda_m} \tag{6.24}$$

Equation (6.23) now becomes:

$$\begin{aligned}
\frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K \left(\alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p}\right)}{\lambda_m \left(1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}\right)} - \lambda_k K \\
\frac{dS}{dt} &= \alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s S \left(\alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p}\right)}{\lambda_m \left(1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}\right)} - \lambda_s S
\end{aligned} \tag{6.25}$$

By numerically solving equations $\frac{dK}{dt} = 0$ and $\frac{dS}{dt} = 0$, we obtain three following fixed points:

$$(K, S) = \{(128.9, 4987.3), (230.6, 5495.1), (5864.6, 1343.2)\}$$

The corresponding eigenvalues for the left-most, intermediate and right-most fixed points as shown on the plane are:

$$\begin{aligned}
(e_k, e_s) = \{&(-0.1076, -0.3478), (0.1073, -0.3434), \\
&(-0.0162 + 0.7798i, -0.0162 + 0.7798i)\}
\end{aligned}$$

As a result, the left-most fixed point is classified as stable while the intermediate and left-most fixed points are saddle and stable focus fixed points, respectively. Figure 6.2 shows the nullcline plane as well as the structure of the fixed points of the 2D adiabatic SynExSlow model. As we can see, the vector field shows that there exists a basin of at-

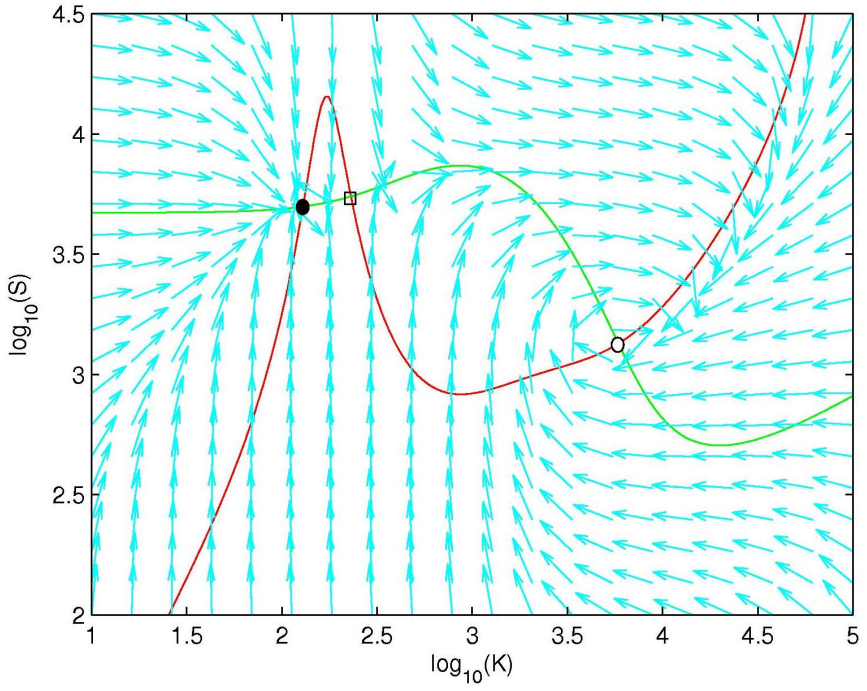


FIGURE 6.2: Nullcline plane of the 2D adiabatic SynExSlow model. The red and green thin lines are the nullclines of ComK and ComS, respectively. The stable fixed point is denoted by a full circle, saddle point by an empty rectangle and the other stable focus point by an empty circle. The arrows show the vector field.

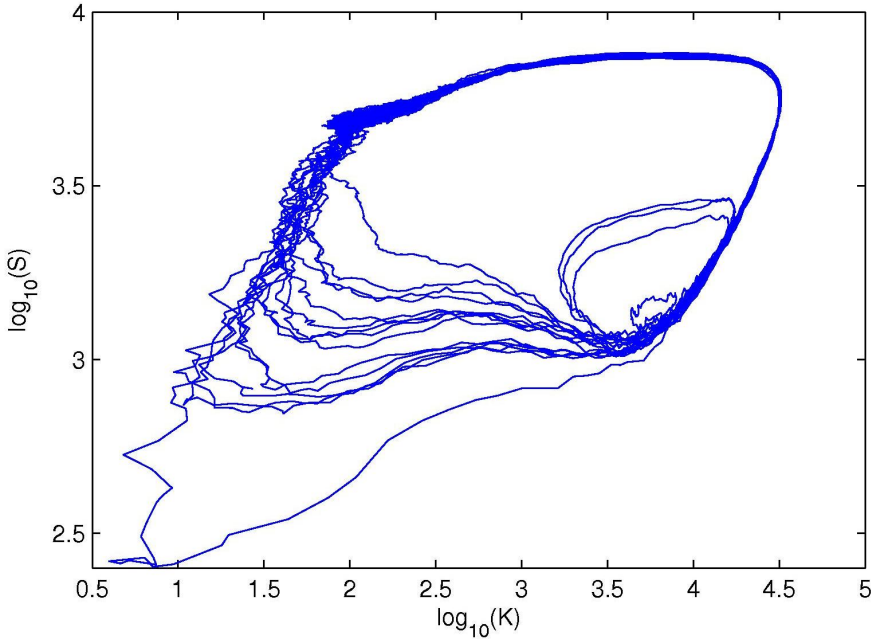
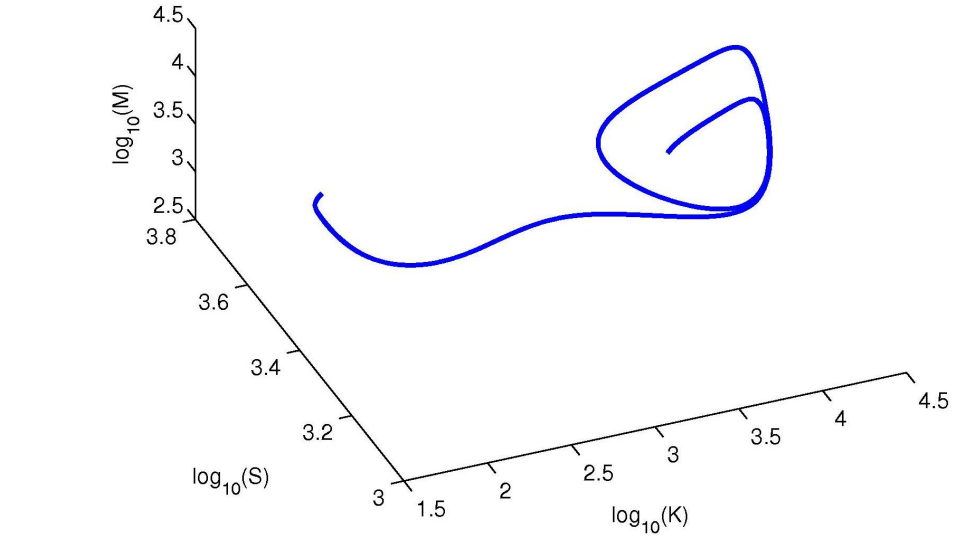
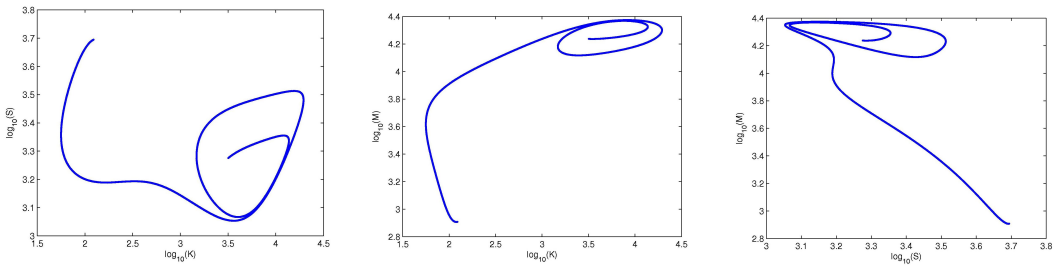


FIGURE 6.3: Trajectories generated by the Langevin simulation.



(a)



(b) (c) (d)

FIGURE 6.4: Trajectory generated by the 3D deterministic model (a), and its projection on logarithmic planes K-S (b), K-M (c) and S-M (d). The numerical initial condition for the integration is $K = 3179$, $S = 1885$, $M = 17266$ (these values are chosen from the Langevin simulation). The existence of intersection point shown in the 2D projections implies that the 3D model cannot be expressed as a 2D system.

traction formed near the stable focus fixed point. This means the oscillations may occur in the case that the trajectories travel close to the fixed point. In fact, the trajectories will be stuck in the attraction region for most of the time since this region is quite large as we can see in Figure 6.2. Additionally, this also shows that these trajectories should either be trapped in the attraction region or move upwards to the other stable fixed point according to the direction of the vector field. However, the Langevin simulation of the 3D SynExSlow model shows that there are only few trajectories being trapped in that region. Although this phenomenon rarely happens (only once in 10000 hour simulation), it has not been observed in Cagatay et al. (2009a). Moreover, for the trajectories which do not fall into the basin of attraction, these trajectories do not follow the vector field but move horizontally along the K-direction before returning back to the other stable fixed point (see Figure 6.3).

On the other hand, Figure 6.4 shows a trajectory generated from the 3D deterministic

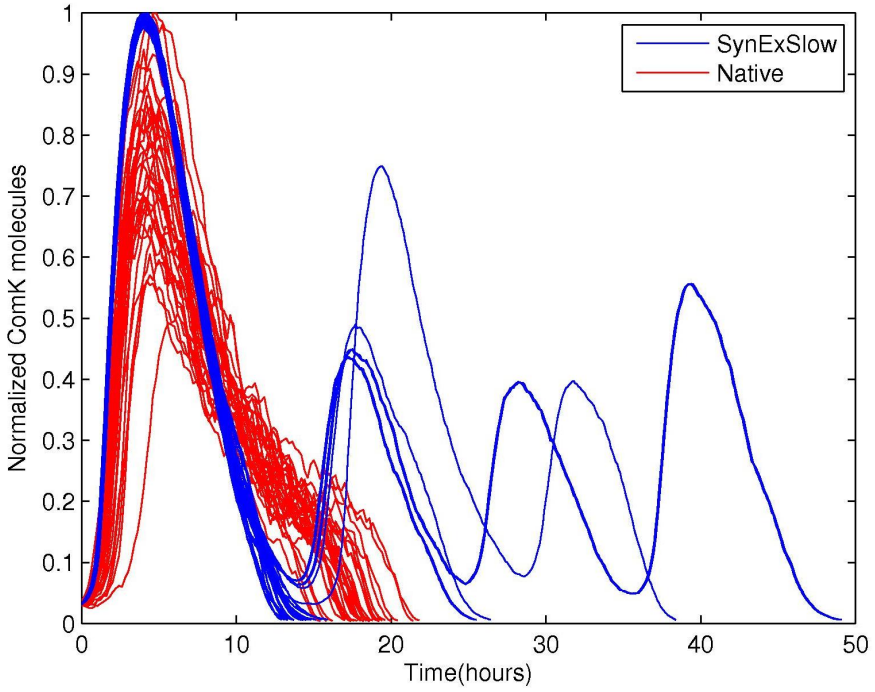


FIGURE 6.5: SynExSlow differs from the Native circuit in competence durations.

model and a projection of this trajectory onto 2D logarithmic planes. As we can see in Figure 6.4, the 2D curve intersects itself due to the existence of attraction basin as shown earlier. At this point, the trajectory can be either trapped in the attraction basin or escape to the other stable fixed point. For this situation, the velocity of the model on a 3D plane (K,S,M) is not uniquely determined by a 2D plane. Thus, the SynExSlow model can not be reduced to a lower-dimensional model.

In addition to our observation, Figure 6.5 shows a comparison between the SynExSlow and the Native circuit in competence durations. In this figure, the competence durations in SynExSlow are less variable than these in the Native circuit; however, small "bumps" caused by oscillations also occur in some competent events in the SynExSlow. Since the Gillespie model fails to be reconstructed, we need to find a way of fixing it so that we can build up a full stochastic model. The method will be detailed in the following section.

6.3 Postulating A Modified CME That Is Consistent

In this section, we try to fix the chemical mechanism where we found $k_{19} < 0$ in order to come up with a correct discrete model which is supposed to describe the same dynamical behaviour as that in the stochastic continuous SynExSlow model. In fact, we can easily fix this problem by changing the values of the model parameters such that $k_{19} > 0$. However, I would show that the new set of parameters does not reproduce the dynamics

$[P_{comK}^{const}]$	$1\,nM$
$[P_{comS}^{const}]$	$1\,nM$
$[P_{mecA}^{const}]$	$1\,nM$
$[P_{comK}]$	$1\,nM$
$[P_{comS}]$	$1\,nM$
$[P_{mecA}]$	$1\,nM$
$[mRNA_{comK}]$	$1000\,nM$
$[mRNA_{comS}]$	$1000\,nM$
$[mRNA_{mecA}]$	$1000\,nM$
$[MecA]$	$300\,nM$
$[MecA_K]$	$100\,nM$
$[MecA_S]$	$100\,nM$
$[ComK]$	$100\,nM$
$[ComS]$	$5000\,nM$
$[X]$	$10\,nM$

TABLE 6.3: Initial conditions.

of the system (the details of this work is presented in section 6.3.2). Alternatively, we can fix this by replacing the reaction

$$\mathrm{MecA_S} \xrightarrow{k_{19}} \mathrm{MecA}$$

by the following reactions

$$\mathrm{MecA_S} \xrightarrow{k_{19}} \mathrm{X}$$
$$\mathrm{MecA + X} \xrightarrow{k_{20}/\Omega} \mathrm{2MecA_S}$$

where X is some complex. Under this assumption, MecA is now being consumed rather than created. Even though this will change the original reaction scheme, it is still worth trying to see if we can reproduce the dynamics. The reaction rate equation of X is described as follows:

$$\frac{d[X]}{dt} = k_{19}[MecA_S] - k_{20}[MecA][X] \tag{6.26}$$

We make k_{20} large enough so the second reaction can be seen as very fast reaction. As a result, its dynamics quickly reaches equilibrium; therefore, we have:

$$\frac{d[X]}{dt} \approx 0 \tag{6.27}$$

yielding:

$$k_{20}[MecA][X] = k_{19}[MecA_S] \tag{6.28}$$

The differential equation for $MecA_S$ is expressed below:

$$\frac{d[MecA_S]}{dt} = k_{18}[ComS][MecA] - k_{-18}[MecA_S] \tag{6.29}$$

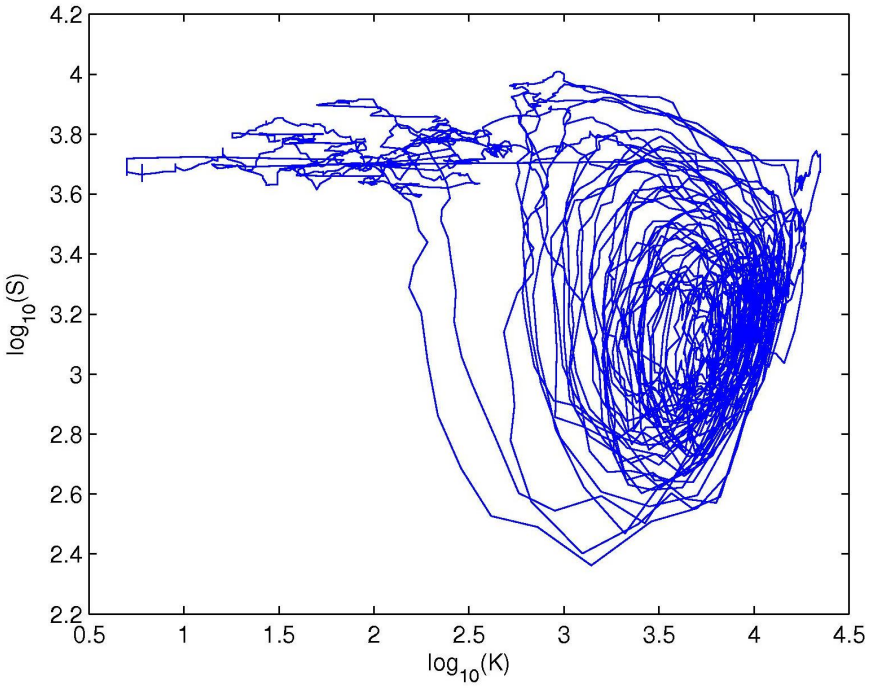


FIGURE 6.6: Sample of trajectories generated from Gillespie simulation.

$$- k_{19}[MecA_S] - k_{15}[MecA_S] + 2k_{20}[X] \tag{6.30}$$

Plugging (6.28) into (6.29), we obtain:

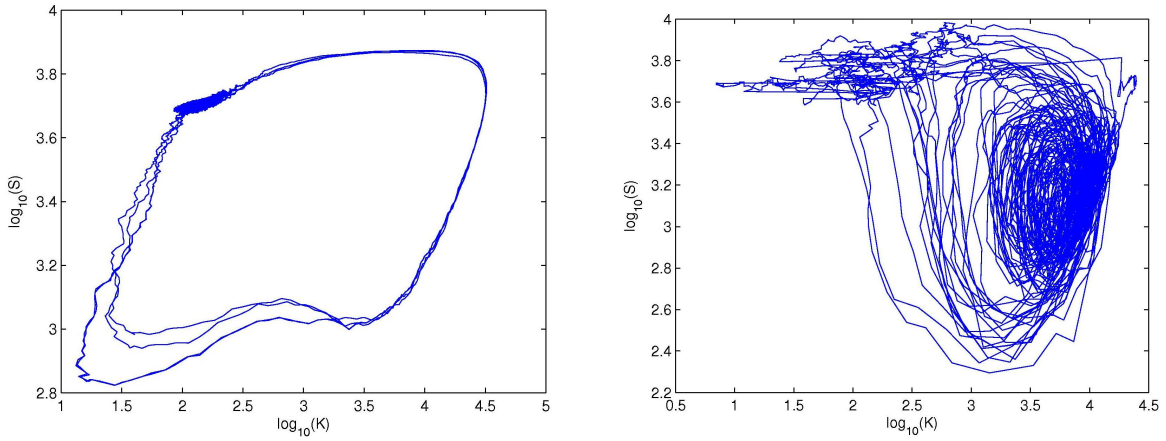
$$\frac{d[MecA_S]}{dt} = k_{18}[ComS][MecA] - k_{-18}[MecA_S] \tag{6.31}$$

$$+ k_{19}[MecA_S] - k_{15}[MecA_S] \tag{6.32}$$

Similarly, by setting $\frac{d[MecA_S]}{dt} \approx 0$ and performing the parameter matching, we have:

$$k_{15} - k_{19} = \delta_s \Gamma_s \Rightarrow k_{19} = k_{15} - \delta_s \Gamma_s = 6 \times 10^{-5} \tag{6.33}$$

As a result, we have built up the system from the biochemical reactions, the system is now supposed to behave in the same way as it does in the stochastic SynExSlow model. Figure 6.6 shows the trajectories which are sampled from running Gillespie simulation with $k_{20} = 1.5$, $\Omega = 1$ (The initial conditions are given in Table 6.3). As we can see in Figure 6.6, the system behaves differently compared to the stochastic SynExSlow model in which the small oscillations occur much more often near the weak stable fixed point. Consequently, the discrepancy between the full Gillespie simulation and the Langevin approximation implies that the Langevin model does not capture the noise distribution that the Gillespie model puts in. In the next experiment, we will see if there is any change in the oscillations when changing the volume of the system. This is based on the fact that the Langevin equation describes the time-evolution of species in terms of molecular numbers, however, the deterministic model explains this in terms of species



(a) (b)

FIGURE 6.7: Sample trajectories generated by Langevin simulation (a), and by Gillespie simulation (b) with $\Omega = 2$.

concentrations. Therefore, it is necessary to describe the Langevin approximation in terms of concentration instead of molecular number. The propensity function of the averages of the concentrations $z_i = x_i/\Omega$ is $a_j(z_i(t)) = \frac{1}{\Omega}a_j(x_i(t))$. As a result, the full Langevin equation can be described as follows:

$$\Omega(x_i(t+\tau) - x_i(t)) = \Omega \sum_{j=1}^{M} \nu_j a_j(x_i)\tau + \sum_{j=1}^{M} \nu_j \sqrt{\Omega}\sqrt{a_j(x_i)}dW_j \tag{6.34}$$

Therefore:

$$x_i(t+\tau) = x_i(t) + \sum_{j=1}^{M} \nu_j a_j(x_i)\tau + \sum_{j=1}^{M} \nu_j \frac{1}{\sqrt{\Omega}}\sqrt{a_j(x_i)}dW_j \tag{6.35}$$

In this equation, we can reduce the noise by increasing the volume Ω. In fact, we will see that the oscillations disappear in the Langevin simulation; however, they still occur in the Gillespie simulation with the same value of volume $\Omega = 2$ (see Figure 6.7). The result obtained from the Langevin simulation agrees with the fact that it is less likely for the trajectories to be trapped in the attractor region if the noise is small. Therefore, there will be no oscillation in this case. However, this does not happen to the Gillespie simulation in the same way as shown in Figure 6.7. It is clear that the inconsistency of those two methods in simulation results shows the ineffectiveness of applying the adiabatic approximation in our system. This has also been observed in the wild-type circuit in the previous chapters. In addition, the SynExSlow model is derived from the Native model which has been proven to poorly describe the dynamical behaviour of the system. Particularly, the assumptions about the fast processes were incorrect leading to the inaccuracy of the adiabatic model in capturing the right systematic behaviour. As a result, the adiabatic SynExSlow model does not provide a correct cellular behaviour of the system.

6.3.1 Ruling Out An Alternative RRE

In fact, we are forced to assume that both $MecA_K$ and $MecA_S$ are degraded in order to derive the deterministic description of the model, where there is no obvious reason for such an assumption. Moreover, an explicit representation of the proteolytic action of $MecA$ was not introduced in the SynEx circuit, which is, however, the step that was modelled in the wild-type with the usual enzymatic mechanism for which the Γ_k and Γ_s were the Michaelis constants. For this reason, it is necessary to keep this mechanism for the explicit description in the SynExSlow circuit. However, another alternative is to remove the degradation of $MecA_K$ and $MecA_S$ in (6.2) so as to keep the description consistent with that in the wild-type. By doing so, there is no change in the deterministic description of ComK and ComS, however, the differential equations for $MecA_K$ and $MecA_S$ now become:

$$\frac{d[MecA_K]}{dt} = k_{16}[ComK][MecA] - k_{-16}[MecA_K] \tag{6.36}$$

$$- k_{17}[MecA_K] \tag{6.37}$$

$$\frac{d[MecA_S]}{dt} = k_{18}[ComS][MecA] - k_{-18}[MecA_S] \tag{6.38}$$

$$- k_{19}[MecA_S] \tag{6.39}$$

Applying the same assumption for the fast processes, $\frac{d[MecA_K]}{dt} \approx 0$ and $\frac{d[MecA_S]}{dt} \approx 0$, we obtain:

$$[MecA_K] = \frac{[MecA][ComK]}{\Gamma_k} \tag{6.40}$$

$$[MecA_S] = \frac{[MecA][ComS]}{\Gamma_s} \tag{6.41}$$

where $\Gamma_k = \frac{k_{17}+k_{-16}}{k_{16}}$, $\Gamma_s = \frac{k_{19}+k_{-18}}{k_{18}}$. Similarly, the differential equations for $mRNA_{mecA}$ and $MecA$ remains unchanged:

$$\begin{aligned}
\frac{d[mRNA_{mecA}]}{dt} &= k_7[P_{mecA}^{const}] + [P_{mecA}]\frac{k_8[ComK]^p}{{k_m}^p + [ComK]^p} \\
&\quad - k_{12}[mRNA_{mecA}] \\
\frac{d[MecA]}{dt} &= k_9[mRNA_{mecA}] - k_{15}[MecA] \\
&\quad - k_{16}[MecA][ComK] + k_{-16}[MecA_K] \\
&\quad + k_{17}[MecA_K] - k_{18}[MecA][ComS] \\
&\quad + k_{-18}[MecA_S] + k_{19}[MecA_S]
\end{aligned} \tag{6.42}$$

By setting $\frac{d[mRNA_{mecA}]}{dt} \approx 0$ and putting back into the equation for $MecA$, we yield:

$$\frac{d[MecA]}{dt} = \frac{k_9 k_7}{k_{12}} + \frac{k_9 k_8}{k_{12}}\frac{[ComK]^p}{{k_m}^p + [ComK]^p} - k_{15}[MecA]$$

$$
\begin{aligned}
&- k_{16}[MecA][ComK] + k_{-16}[MecA_K] \\
&+ k_{17}[MecA_K] - k_{18}[MecA][ComS] \\
&+ k_{-18}[MecA_S] + k_{19}[MecA_S]
\end{aligned} \tag{6.43}
$$

From (6.43) and (6.42), we obtain:

$$
\frac{d[MecA]}{dt} = \frac{k_9 k_7}{k_{12}} + \frac{k_9 k_8}{k_{12}} \frac{[ComK]^p}{{k_m}^p + [ComK]^p} - k_{15}[MecA] \tag{6.44}
$$

We now use notation M for the total concentration of MecA, while K, S for the

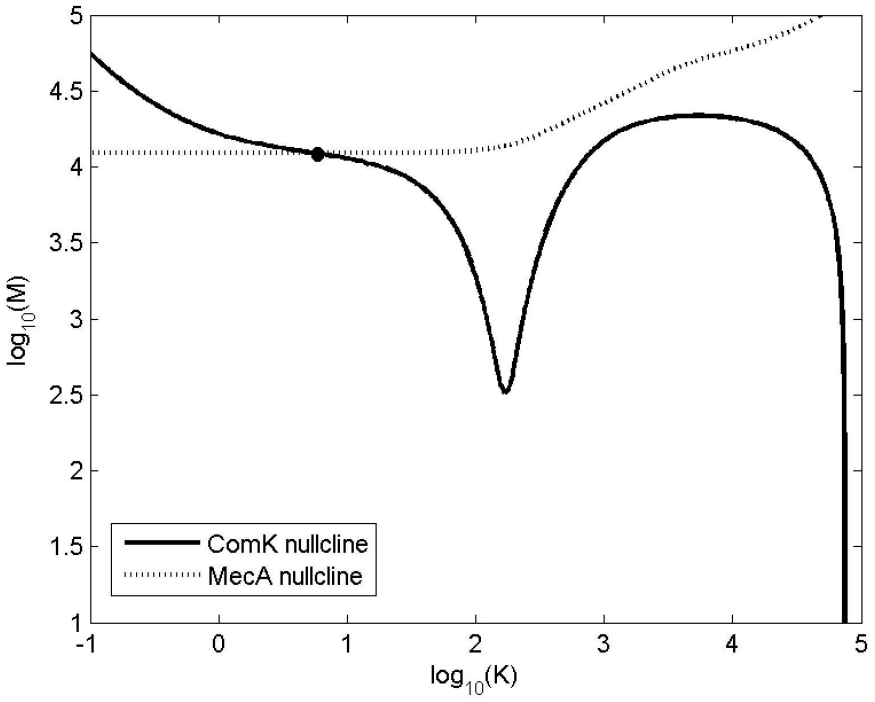


FIGURE 6.8: Nullclines of ComK and MecA in the new SynExSlow model. Those nullclines intersect at only one stable fixed point (full circle).

concentration of ComK and ComS. We have $\frac{dM}{dt} = \frac{d[MecA]}{dt}$, and we also have

$$
[MecA] = \frac{M}{1 + \frac{[ComK]}{\Gamma_k} + \frac{[ComS]}{\Gamma_s}} \tag{6.45}
$$

Plugging (6.45) into (6.44), we obtain

$$
\begin{aligned}
\frac{dM}{dt} &= \frac{k_9 k_7}{k_{12}} + \frac{k_9 k_8}{k_{12}} \frac{[ComK]^p}{{k_m}^p + [ComK]^p} - k_{15} \frac{M}{1 + \frac{[ComK]}{\Gamma_k} + \frac{[ComS]}{\Gamma_s}} \\
&= \alpha_m + \frac{\beta_m [ComK]^p}{{k_m}^p + [ComK]^p} - \lambda_m \frac{M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}}
\end{aligned}
$$

Finally, we end up with the following deterministic description for the 3D SynExSlow:

$$\begin{aligned}
\frac{dM}{dt} &= \alpha_m + \frac{\beta_m K^p}{{k_m}^p + K^p} - \lambda_m \frac{M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} \\
\frac{dK}{dt} &= \alpha_k + \frac{\beta_k K^n}{{k_k}^n + K^n} - \frac{\delta_k K M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_k K \\
\frac{dS}{dt} &= \alpha_s + \frac{\beta_s K^n}{{k_s}^n + K^n} - \frac{\delta_s S M}{1 + \frac{K}{\Gamma_k} + \frac{S}{\Gamma_s}} - \lambda_s S
\end{aligned} \tag{6.46}$$

where $\delta_k = k_{17}/\Gamma_k$ and $\delta_s = k_{19}/\Gamma_s$. However, the new SynExSlow model is not a correct model since it has only one stable fixed point (Figure 6.8).

6.3.2 Ruling Out An Alternative Set of Parameters

As mentioned in section 6.3, we can easily fixed the issue where $k_{19} < 0$ by changing the model parameters. Particularly, we need $k_{19} = \delta_s \Gamma_s - k_{15} > 0$; therefore, we obtain either $\Gamma_s > k_{15}/\delta_s = 50$ or $\delta_s > k_{15}/\Gamma_s = 5 \times 10^{-6}$. It would be easier to keep Γ_s as it is while changing δ_s since the nullcline of ComK will remain the same while the nullcline of ComS varies. Assuming that we choose $\delta_s = 6 \times 10^{-6}$, then $k_{19} = \delta_s \Gamma_s - k_{15} = 2 \times 10^{-5}$. On the other hand, since (6.20), we have $k_{-18} = k_{18}\Gamma_s - k_{15} - k_{19} > 0$; therefore, $k_{18} > \frac{k_{15}+k_{19}}{\Gamma_s} = 6 \times 10^{-6}$. Here, we can take $k_{18} = 7 \times 10^{-6}$.

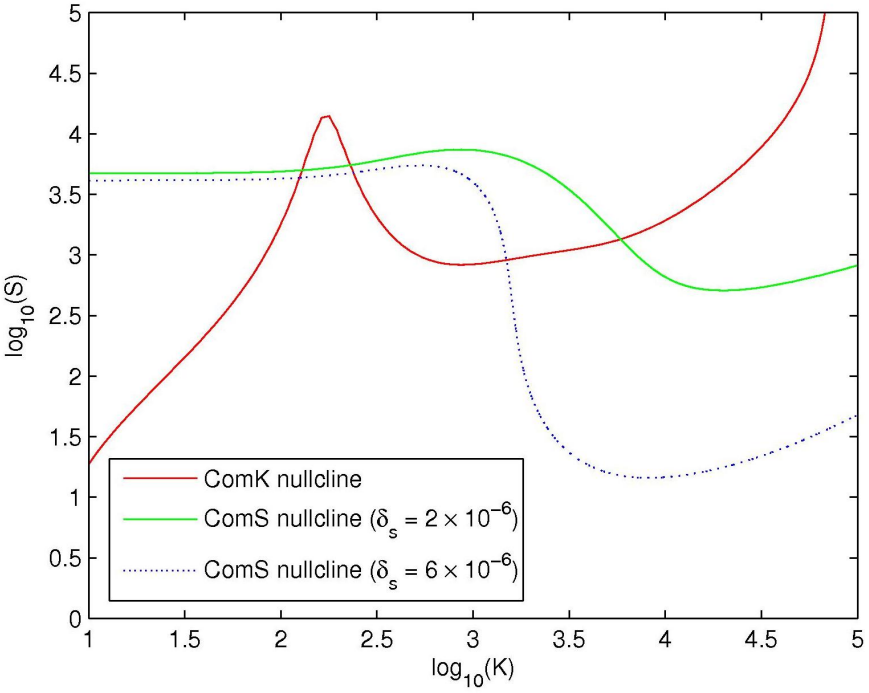
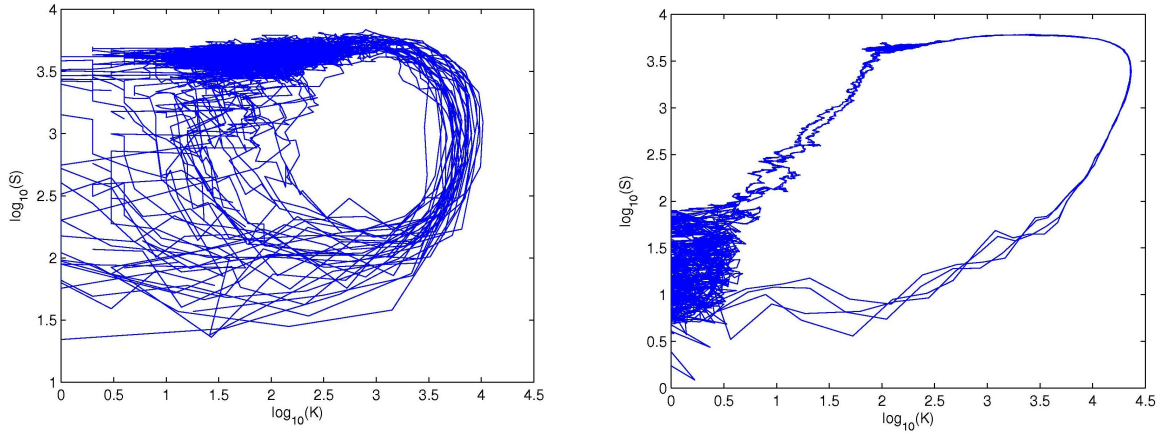


FIGURE 6.9: The variation of the nullclines under parameter (δ_s) changes. The new nullcline of ComS ($\delta_s = 6 \times 10^{-6}$) moves downward compared to the original one ($\delta_s = 2 \times 10^{-6}$), the right-most fixed point is therefore shifted to the left.



(a) (b)

FIGURE 6.10: Trajectories generated from the Gillespie simulation (a), and Langevin simulation (b).

Figure 6.9 shows the variation of the nullclines under the change of parameter δ_s. We now see if the new set of parameters reproduce the dynamics of the system. In order to verify this, I do Gillespie simulation for the full system (the initial condition is given in Table 6.3) and then compare with that in the Langevin simulation described in (6.22). In fact, trajectories generated from the Gillespie simulation are very different from that in the Langevin simulation (see Figure 6.10). In Figure 6.10, the Gillespie simulation is pretty much noisier than that in the Langevin simulation; therefore, trajectories generated from the Langevin simulation look smoother. This means the Langevin model does not capture the noise distribution that the Gillespie model puts in. In other words, the new set of parameters does not help us reproduce the dynamics of the system.

6.4 Summary

In this chapter, I have performed the same analysis on the synthetic circuit SynExSlow regarding to model reduction. The aim of this analysis is to generalize our method to excitable circuits. However, my study has shown the challenges in describing the dynamical behaviour of bacteria in the SynExSlow model. These challenges come from the fact that there is a gap between simulating the cellular behaviour using Gillespie simulation and the stochastic Langevin approximation. In fact, the model failed in describing the real behaviour of the system as it could not be built up from a set of biochemical reactions. On the other hand, the Langevin approximation still does not capture the right noise in the Gillespie simulation though the circuit can be successfully re-constructed. Moreover, the presence of small oscillations which have not been observed in the experimental data mentioned in Cagatay et al. (2009a), leads to the impossibility of doing model reduction. Consequently, these results strongly prove that the model reduction can not be done by using adiabatic approximation alone, but needs to introduce a better

solution so that the dynamics of the system can be well approximated.

Chapter 7

Conclusions

7.1 Results and Evaluation

Genetic circuit of competence helps us understand the sophisticated biochemical reactions as well as the cellular mechanisms in bacteria *B. subtilis*. The discrete stochastic model which has been used in the genetic circuit can simulate the system, but provides limited insight. A deterministic continuous approximation gives deeper insight into the system through the study of fixed points, but missed out noise induced dynamics which is very important. Although the 2D adiabatic model is simple enough to understand the cellular behaviour near the steady state, it is limited in providing correct dynamics in the whole phase. Thus, reducing a very complicated high-dimensional model to a much simpler low-dimensional model while preserving the right dynamics of the system is critical to understanding cellular behaviour. We can then apply tools such as Fokker-Planck equation to analyze the stochastic dynamic system which is characterized by its time-parameterized probability density function (PDF). This also allows us to be able to compute the initialization probability per unit time of the cell being in competent state explicitly.

The thesis has shown experimental simulations which reproduced the results presented by Süel et al. (2007) including trajectories, vector field, initialization probability, competence duration, etc. However, it also shows the discrepancy in competence durations between the simulation result computed by running Gillespie algorithm and that provided in the paper. In particular, it is explained that the dynamics of proteins during the excitable state is actually faster than that discussed in the paper; therefore, the competence duration cells spend during the competent state should be shorter. Moreover, our simulation has also shown that the noise basically does not impact on the competence duration; however, the noise near the switching state can significantly effect the initialization probability.

Apparently, the 2D naive adiabatic model meets lots of limitations as it describes the

dynamics of a noise-induced system. For this reason, I have tried to come up with a simple approximate model which can describe the dynamics of the system better. As a result, I have built up a 2D approximate model using slow invariant manifold technique and an iterative procedure which is easy to implement. The 2D approximate model is verified to have preserved the fixed points and their stability in the original system. Additionally, the simulation results showed the right behaviour in the excitable regime where the competence duration is computed to be about 10 hours which is the same as that in the full system. However, it is limited in describing the right behaviour at the transient where cells spend a short period of time before getting back to steady state. This is because the dynamics in mRNA molecules which were assumed to be very fast had not been that fast. In other words, the time scales of the dynamics are not completely separated. Consequently, applying the adiabatic approximation in this case did not work properly. Moreover, the singular perturbation method used to capture the fast processes did not improve the approximation either.

I also showed that the competence mechanism relies on the fluctuation of very small number of mRNA molecules which is amplified through a positive feedback loop. The role of feedback mechanism has also been studied as looking at the stochastic noise in a single gene regulatory network, Tao et al. (2007) showed that a gene with feedback regulation will have different total noise in the number of proteins compared to that with the same average of protein molecules. Clearly, the effect of feedback regulation mechanism on gene expression is critical to the dynamical behaviour of cells; therefore, it has to be taken into account as doing the model reduction for any stochastic continuous model. In our case, the simulation results have also shown that it is impossible to reduce a Langevin model by using adiabatic approximation alone. This is not only because of the fluctuations in proteins being ignored but also because of the impact of the positive feedback which is known as essential for the bistability (Maamar and Dubnau, 2005) having been missed out. On the other hand, the bistable genetic circuit is believed to be very sensitive to the switching behaviour (Mehta et al., 2008). The simulation results showed that a slight difference at the tail of the probability distribution among approximate models may result in large variation in the number of cells entering the competent state. Particularly, the Langevin approximate models produce much higher initialization probability for cells to become competent than that in the Gillespie model as a consequence of the exponential sensitivity of switching state.

In supplement to these results, I analyzed alternative regulatory networks SynEx and SynExSlow which exhibit a similar behaviour as observed in the wild-type circuit. Our simulations have shown the ODE description from which dynamical behaviour was derived does not follow a chemical reaction description. Moreover, there is a discrepancy between the experimental result claimed by Cagatay et al. (2009a) and the simulation data. Particularly, our simulation showed small oscillations occurring near the unstable fixed point at the excitable regime; however, these have not been observed in their

experimental result. As a result, this implies the impossibility of doing model reduction.

To sum up, I have presented an analysis of a competence genetic circuit which has been typically used for researching cellular behaviour in bacteria *Bacillus subtilis*. Not only have we reproduced the results in the paper but also provided a correct way of observing the dynamical behaviour of the system. There are three important results we have found through out the thesis. Firstly, the fluctuations in mRNA strongly contribute to the total noise of the system, driving the system to the excitable state. As a result, these fluctuations are unable to be ignored when doing the model reduction. Secondly, the positive feedback which has not actually been captured in the reduction approaches does impact on the noise of protein. This, as a result, leads to an inefficiency in our approximation methods. Thirdly, the tail of the probability distribution is very sensitive to the initialization probability which describes how often cells enter the competent state. These results are vital to better understanding the real behaviour in genetic circuit. In addition, I have also shown that, it is the very small number of mRNA as well as the impact of positive feedback loop that cause the failure of the Langevin simulation. On the other hand, it is very important to find out about the Langevin equation that the variance of protein is proportional to square of its molecular number. This result allows us to construct a reduced model which better approximate the dynamical behaviour of the original model by putting an empirical noise back into the Langevin equation. We then can solve the Fokker-Planck for the stochastic model in order to produce a complete histogram of the species in terms of probability distribution. The empirical noise can be tuned such that the probability distribution obtained from the Fokker-Planck best fits that computed from the original model.

7.2 Future Work

In this thesis, we have presented an analysis of a competence genetic circuit which exhibits the natural behaviour of bacterium. A part from the simulation results obtained, however, the problem of dimensionality reduction has not yet been solved. In fact, a completion of solution to the problem is critical to solving a variety of similar problems. In particular, this method can be used to deal with some other complex genetic circuits and make them simpler for analysis. On the other hand, our research has also showed that there is a gap between the Gillespie and Langevin simulations in describing the cellular behaviour. Moreover, the initialization probability obtained from the Langevin simulation is much larger than that computed in the Gillespie simulation. In fact, the Langevin approximation does not work effectively when the species populations are too small; therefore, it is necessary to set out a solution to the Langevin method for this particular case.

Even though there are challenges in modelling gene regulation network, we have found

useful evidences which help us understand the wild-type genetic circuit better. Based on these findings, we have come up with a relatively good approximate model which can give us deep insights into the dynamics of the system. Moreover, this is an important starting point to the generalisability of our method to genetic circuits that generate excitable dynamics. For this reason, we have already submitted a paper to show these results.

In additon, I showed the sensitivity of the switching behaviour at the tail of the probability distribution which was also observed by Mehta et al. (2008). The simulation results also showed the significant contribution of the fluctuations in mRNA to the total noise of the system. As a result, ignoring these fluctuations may lead to a wrong dynamical behaviour. On the other hand, the size of fluctuation of protein in the chemical Langevin equations has been found to be proportional to the mean of protein. It means that the fluctuation itself is much larger than it should be due to the effect of the positive feedback scheme. Although I have provided evidences for this observation, the question of how the positive feedback loop impacts on the fluctuation, however, still remains unanswered. Thus, it should be nice if we can mathematically describe this relationship in order to better understand the system. Consequently, we may come up with a better solution to the model reduction in which the dynamical behaviour can be well approximated.

Appendix A

Appendix

A.1 Mechanism

A.1.1 Transcription

Transcription is the first stage of gene expression in which mRNA is synthesized from a DNA template. At first, RNA polymerase binds to a specific base sequence in the DNA called a promoter which is close to the start of the coding region of a gene. The RNA polymerase then unwinds the DNA before a strand of gene is copied to RNA (Andrey et al., 2006). After that, the RNA polymerase then adds more complementary strands of RNA in order to create a message which is called mRNA (Robinson and van Oijen, 2013). mRNA contains regions which are not used in translating into proteins, these regions are called introns and will be removed from mRNA so as to form mature mRNA, which is able to leave the nucleus through pores and go into cytoplasm. In other words, this process is basically to write down a message which is contained in DNA preparing for the next stage known as translation. Bacteria, however, do not have a distinct nucleus so there is no barrier to immediate translations; therefore, the transcription and translation occur simultaneously in this case (Ralston, 2008).

A.1.2 Translation

During this process, the mRNA moved to the cytoplasm is decoded or translated in order to produce the correct order of amino acids in a protein. In fact, mRNA after entering the cytoplasm will become associated with ribosomes which are a combination of rRNA and proteins. According to that, every three letters in mRNA codes for one amino acid, and each initiator tRNA that pairs up with the mRNA codons, carries a specific amino acid down to the ribosomes and drop that off to the growing protein chain, producing protein base by base. After dropping that down, the tRNA naturally goes inside the

cytoplasm to gather another amino acid and this process is iterated until the ribosomes hits the stop sequences encoded as UAA, UGA or UAG (Alberts, 2002).

A.2 Regulation of Gene Expression

A.2.1 Transcription Regulation

Transcriptional regulation is the way a cell controls how often a given gene is transcribed by making conditions for transcription initiation more favourable or less favourable. In eukaryotic cells, transcription is controlled by proteins that bind to specific regulatory sequences and modulate the activity of RNA polymerase (Geoffrey and Cooper, 2000). In this process, transcription factors, which are proteins that bind to DNA in a sequence specific manner to regulate transcription, alter the rate at which transcripts are produced. The transcription factors regulate transcription by either enhancing or preventing the recruitment and binding of the RNA polymerase to the promoter of the gene (Carey et al., 1999; Locker, 2000). In biology, transcription factors play a critical role in development and differentiation of organisms. They can act in many different biological contexts during development and can regulate many different gene programs in different organisms (Zeitlinger and Stark, 2010).

A.2.2 Post-transcriptional Regulation

Post-transcriptional regulation of gene expression also determines how much mRNA is translated into proteins. Cells may do it by several ways including mRNA processing (polyadenylation, capping, and splicing), mRNA export and localization, mRNA decay, and mRNA translation (Dubnau, 1991; Day and Tuite, 1998). In this process, moreover, RNA binding proteins (RBPs) play a critical role in the development of mRNA regulation and protein abundance. In addition, RNAs may contain more than one RBP-binding site that is associated with multiple RNAs to be able to form ribonucleoprotein (RNP) complexes which are involved in activities of cell metabolism such as DNA replication, expression of histone genes, regulation of transcription and translational control (Cagatay et al., 2009b).

A.2.3 Translational Regulation

Translational regulation is about the control of the levels of protein synthesized from its mRNA. The mechanisms are centred on the control of ribosome recruitment for the initiation codon. They also are involved in the modulation of the elongation or termination of protein synthesis. Basically, translational regulation includes specific

RNA secondary structures on the mRNA. In eukaryotic cells, the translational control is critical for gene regulation during nutrient deprivation and stress, development and differentiation, nervous system function, aging, and disease (Sonenberg and Hinnebusch, 2009).

A.2.4 Protein Degradation

The level of protein concentration is both dependent on its rate of synthesis and its rate of degradation; therefore, the regulation of protein degradation represents a potential mechanism for modulating gene expression (Callis, 1995). In this process, proteasomes which are very large protein complexes play a critical role in regulation mechanism by which cells may regulate the concentration of particular proteins and degrade misfolded proteins. Moreover, proteins are also labeled for degradation with a small protein called ubiquitin which has been found in almost all tissues of eukaryotic organisms but not in bacteria.

A.3 Hill Equation

In multiple-binding-site mechanism such as the activity of enzymes, the binding of a molecule of substrate to one site may influence (activate or inhibit) the binding of another of substrate to the second site. In order to quantify this effect, the Hill equation (Hill, 1910, 1913) was introduced and has been widely applied in biochemistry. To understand how the Hill equation works, we first take an example of a chemical mechanism where enzyme E convert other molecules called S into products P. Here, we assume that E has two binding sites, and the substrate S can equally bind to one of the binding sites to form a complex C. The chemical reactions are therefore given as follows:

$$\begin{aligned} \mathrm{S} + \mathrm{E} &\underset{c_{-1}}{\overset{c_1}{\rightleftharpoons}} \mathrm{C}_1 \xrightarrow{c_2} \mathrm{E} + \mathrm{P} \\ \mathrm{S} + \mathrm{C}_1 &\underset{c_{-3}}{\overset{c_3}{\rightleftharpoons}} \mathrm{C}_2 \xrightarrow{c_4} \mathrm{C}_1 + \mathrm{P} \end{aligned} \tag{A.1}$$

Let $E_T = E + C_1 + C_2$. This quantity remains constant since all reactions involving E or its complexes conserves this sum. Notice that enzyme E has two binding sites; therefore we should have two different forms of C_1 and C_2. The differential equation for S, C_1,

C_2 and P are described as follows:

$$\begin{aligned}
\frac{dE}{dt} &= -c_1SE + (c_{-1} + c_2)C_1 \\
\frac{dS}{dt} &= -c_1SE + c_{-1}C_1 - c_3SC_1 + c_{-3}C_2 \\
\frac{dC_1}{dt} &= c_1SE - (c_{-1} + c_2)C_1 - c_3SC_1 + (c_{-3} + c_4)C_2 \\
\frac{dC_2}{dt} &= c_3SC_1 - (c_{-3} + c_4)C_2 \\
\frac{dP}{dt} &= c_2C_1 + c_4C_2
\end{aligned} \tag{A.2}$$

The rate of generating P is given by $v = \frac{dP}{dt} = c_2C_1 + c_4C_2$. At the steady state, we set

$$\frac{dC_1}{dt} = \frac{dC_2}{dt} = 0 \tag{A.3}$$

and define

$$\begin{aligned}
K_1 &= \frac{c_{-1} + c_2}{c_1} \\
K_2 &= \frac{c_{-3} + c_4}{c_3}
\end{aligned} \tag{A.4}$$

we obtain:

$$\begin{aligned}
C_1 &= \frac{SE}{K_1} \\
C_2 &= \frac{SC_1}{K_2} = \frac{S^2E}{K_1K_2}
\end{aligned} \tag{A.5}$$

therefore, $E_T = E + C_1 + C_2 = E + \frac{SE}{K_1} + \frac{S^2E}{K_1K_2} = E\left(\frac{S}{K_1} + \frac{S^2}{K_1K_2}\right)$, this yields

$$E = \frac{E_T}{1 + \frac{S}{K_1} + \frac{S^2}{K_1K_2}} \tag{A.6}$$

Assuming that the unbinding rates are independent, this means $c_2 = c_4 = c_p$, $c_{-1} = c_{-3} = c$, we obtain

$$v = c_2C_1 + c_4C_2 = c_pE(\frac{S}{K_1} + \frac{S^2}{K_1K_2}) \tag{A.7}$$

From (A.6) and (A.7) we have:

$$v = \frac{E_Tc_p\left(\frac{S}{K_1} + \frac{S^2}{K_1K_2}\right)}{1 + \frac{S}{K_1} + \frac{S^2}{K_1K_2}} \tag{A.8}$$

We now assume that the binding of substrate S to binding site of enzyme E activates the binding of S to the complex C_1, this means $c_3 \gg c_1$. As a result, $\frac{K_2}{K_1} = \frac{c_1}{c_3} = \alpha \ll 1$,

hence

$$v = \frac{E_T c_p \left(\frac{S}{K_1} + \frac{S^2}{\alpha K_1^2}\right)}{1 + \frac{S}{K_1} + \frac{S^2}{\alpha K_1^2}} \tag{A.9}$$

For $\frac{S}{K_1} = \alpha \frac{C_2}{C_1} \ll 1$, this means $S \ll K_1$, we have

$$\begin{aligned} v &\approx \frac{V_{max} \frac{S^2}{\alpha K_1^2}}{1 + \frac{S^2}{\alpha K_1^2}} \\ v &\approx \frac{V_{max} S^2}{K + S^2} \end{aligned} \tag{A.10}$$

where $V_{max} = E_T c_p$, $K = \alpha K_1^2$. The last equation, called Hill equation, has sigmoidal shape and in general case, this equation is expressed as follows:

$$v = \frac{V_{max} S^n}{K^n + S^n} \tag{A.11}$$

where n is a Hill coefficient but not the number of binding sites. In fact, n is always smaller the number of binding sites.

A.4 Linearized Approximation

The linearized approximation is applied to explore the microscopic fluctuation near the steady state of a stochastic process. In particular, this method allows us to mathematically describe the time-evolution of the covariances in terms of the parameters in the rate equation and the stochastic fluctuation around the steady state. Firstly, we assume that $\mathbf{X}(t)$ is a vector of the number of molecules X_i according to each chemical species $S_i (i = 1, 2..., N)$ in the system, then the time-evolution description of the joint probability distribution of all species is given by the following Chemical Master Equation (CME):

$$\frac{\partial}{\partial t} P(\mathbf{X}, t) = \sum_{j=1}^{M} \left[P\left(\mathbf{X} - \boldsymbol{\nu}_j, t\right) a_j \left(\mathbf{X} - \boldsymbol{\nu}_j\right) - P\left(\mathbf{X}, t\right) a_j \left(\mathbf{X}\right) \right] \tag{A.12}$$

where the propensity function a_j gives the probability that one reaction $\mathbf{R}_j \, (1 \leq j \leq M)$ will occur in the next time interval $[t, t + dt)$ in volume Ω. The stoichiometry ν_{ij} represents the changing amount in the number of S_i molecules caused by the reaction $\mathbf{R}_j$. By multiplying Equation (A.12) by $\mathbf{X} = \mathbf{X} - \boldsymbol{\nu}_j + \boldsymbol{\nu}_j$ and summing over all $\mathbf{X}$ and noting that $\langle \mathbf{X} \rangle_t = \sum_X \mathbf{X}\left(t\right) P\left(\mathbf{X}, t\right)$, we get the following expression:

$$\langle X_i(t + \Delta t) \rangle = \langle X_i(t) \rangle + \Delta t \sum_{j=1}^{M} \langle \nu_{ij} a_j(\mathbf{X} - \boldsymbol{\nu}_j) \rangle$$

$$+\Delta t\left\{\sum_{X}\sum_{j=1}^{M}\left(P\left(\mathbf{X}-\boldsymbol{\nu}_j,t\right)a_j\left(\mathbf{X}-\boldsymbol{\nu}_j\right)-P\left(\mathbf{X},t\right)a_j\left(\mathbf{X}\right)\right)\right\} \tag{A.13}$$

The last term in $\{.\}$ vanishes by defining $\mathbf{Y}=\mathbf{X}-\boldsymbol{\nu}$. Thus, we yield:

$$\frac{d\left\langle X_i\right\rangle}{dt}=\sum_{j=1}^{M}\left\langle \nu_{ij}a_j(\mathbf{X})\right\rangle \tag{A.14}$$

Similarly, by multiplying Equation (A.12) by $\left(\mathbf{X}-\langle\mathbf{X}\rangle\right)\left(\mathbf{X}^{\intercal}-\langle\mathbf{X}^{\intercal}\rangle\right)$ and shifting variables in the sum over $\mathbf{X}$, we get the following expression for the covariances:

$$\frac{dCov\left(X_i,X_j\right)}{dt}=\sum_{k=1}^{M}\left\langle\left(X_i-\langle X_i\rangle\right)\nu_{jk}a_k(\mathbf{X})\right\rangle+\left\langle\left(X_j-\langle X_j\rangle\right)\nu_{ik}a_k(\mathbf{X})\right\rangle+\left\langle\nu_{ik}\nu_{jk}a_k(\mathbf{X})\right\rangle \tag{A.15}$$

where $Cov\left(X_i,X_j\right)=\left\langle\left(X_i-\langle X_i\rangle\right)\left(X_j-\langle X_j\rangle\right)\right\rangle$. By doing a linearized approximation of the propensities around the steady state $\mathbf{X}^*$, we yield the time-evolution of the covariances $\mathbf{C}_{ij}:=Cov\left(X_i,X_j\right)$ as follows (see Avi and Ben (2012)):

$$\frac{d}{dt}\mathbf{C}=\mathbf{JC}+\mathbf{CJ}^{\intercal}+\mathbf{BB}^{\intercal} \tag{A.16}$$

where $\mathbf{J}$ is the Jacobian matrix computed at the fixed point $\mathbf{X}^*$, $\mathbf{B}$ is the diffusion matrix which can be derived from the Langevin equation. At the steady state, the covariance matrix $\mathbf{C}$ can be obtained by solving the Lyapunov equation for the steady state $\left(\frac{d}{dt}\mathbf{C}=0\right)$:

$$\mathbf{JC}+\mathbf{CJ}^{\intercal}+\mathbf{BB}^{\intercal}=0 \tag{A.17}$$

In our 7D system, the diffusion matrix $\mathbf{B}$ is defined as follows:

$$B=\begin{array}{c}\\(K)\\(S)\\(A)\\(M_K)\\(M_S)\\(R_K)\\(R_S)\end{array}\begin{array}{c}\begin{array}{cccccccccccccccc}\end{array}\\\begin{pmatrix}0&0&\sqrt{k_3R_K}&0&0&0&0&-\sqrt{k_8K}&0&0&-\sqrt{k_{11}KA}&\sqrt{k_{-11}M_K}&0&0&0&0\\0&0&0&0&0&\sqrt{k_6R_S}&0&0&0&-\sqrt{k_{10}S}&0&0&-\sqrt{k_{13}AS}&\sqrt{k_{-13}M_S}&0&0\\0&0&0&0&0&0&0&0&0&0&-\sqrt{k_{11}AK}&\sqrt{k_{-11}M_K}&-\sqrt{k_{13}AS}&\sqrt{k_{-13}M_S}&\sqrt{k_{12}M_K}&\sqrt{k_{14}M_S}\\0&0&0&0&0&0&0&0&0&0&\sqrt{k_{11}AK}&-\sqrt{k_{-11}M_K}&0&0&-\sqrt{k_{12}M_K}&0\\0&0&0&0&0&0&0&0&0&0&0&0&\sqrt{k_{13}AS}&-\sqrt{k_{-13}M_S}&0&-\sqrt{k_{14}M_S}\\\sqrt{k_1}&\sqrt{\frac{k_2K^n}{k_k^n+K^n}}&0&0&0&0&-\sqrt{k_7R_K}&0&0&0&0&0&0&0&0&0\\0&0&0&\sqrt{k_4}&\sqrt{\frac{k_5}{1+\left(\frac{K}{k_s}\right)^p}}&0&0&0&-\sqrt{k_9R_S}&0&0&0&0&0&0&0\end{pmatrix}\end{array}$$

where K, S, A, M_K, M_S, R_K, R_S are denoted for $[ComK]$, $[ComS]$, $[MecA]$, $[MecA_K]$, $[MecA_S]$, $[mRNA_{comK}]$ and $[mRNA_{comS}]$. Since we have the conservation law for the complex protease, we therefore can eliminate one out of three rows corresponding to $MecA$, $MecA_K$ and $MecA_S$. For example, by removing the row for M_S, we get the corresponding diffusion matrix for the six remaining variables:

$$B=\begin{array}{c}\\(K)\\(S)\\(A)\\(M_K)\\(R_K)\\(R_S)\end{array}\begin{array}{c}\begin{array}{cccccccccccccccc}\end{array}\\\begin{pmatrix}0&0&\sqrt{k_3R_K}&0&0&0&0&-\sqrt{k_8K}&0&0&-\sqrt{k_{11}KA}&\sqrt{k_{-11}M_K}&0&0&0&0\\0&0&0&0&0&\sqrt{k_6R_S}&0&0&0&-\sqrt{k_{10}S}&0&0&-\sqrt{k_{13}AS}&\sqrt{k_{-13}M_S}&0&0\\0&0&0&0&0&0&0&0&0&0&-\sqrt{k_{11}AK}&\sqrt{k_{-11}M_K}&-\sqrt{k_{13}AS}&\sqrt{k_{-13}M_S}&\sqrt{k_{12}M_K}&\sqrt{k_{14}M_S}\\0&0&0&0&0&0&0&0&0&0&\sqrt{k_{11}AK}&-\sqrt{k_{-11}M_K}&0&0&-\sqrt{k_{12}M_K}&0\\\sqrt{k_1}&\sqrt{\frac{k_2K^n}{k_k^n+K^n}}&0&0&0&0&-\sqrt{k_7R_K}&0&0&0&0&0&0&0&0&0\\0&0&0&\sqrt{k_4}&\sqrt{\frac{k_5}{1+\left(\frac{K}{k_s}\right)^p}}&0&0&0&-\sqrt{k_9R_S}&0&0&0&0&0&0&0\end{pmatrix}\end{array}$$

In according to that, the Jacobian matrix **A** is expressed as below:

$$J = \begin{array}{c} \\ (K) \\ (S) \\ (A) \\ (M_K) \\ (R_K) \\ (R_S) \end{array} \begin{array}{c} \begin{array}{cccccc} (K) & (S) & (A) & (M_K) & (R_K) & (R_S) \end{array} \\ \begin{pmatrix} -k_8 - k_{11}A & 0 & -k_{11}K & k_{-11} & k_3 & 0 \\ 0 & -k_{10} - k_{13}A & -k_{13}S & 0 & 0 & k_6 \\ -k_{11}A & -k_{13}A & -k_{11}K - k_{13}S & k_{12} + k_{-11} & 0 & 0 \\ k_{11}A & 0 & k_{11}K & -k_{12} - k_{-11} & 0 & 0 \\ \frac{K^{n-1}k_2{k_k}^n n}{(K^n + {k_k}^n)^2} & 0 & 0 & 0 & -k_7 & 0 \\ -\frac{k_5\left(\frac{K}{k_s}\right)^p p}{K\left(1+\left(\frac{K}{k_s}\right)^p\right)^2} & 0 & 0 & 0 & 0 & -k_9 \end{pmatrix} \end{array}$$

A.5 Ito's Lemma

The Ito's lemma is used to find a stochastic process of a time-dependent function. In particular, assuming that we have the following stochastic process:

$$dx = a\,(x,t)\,dt + b\,(x,t)\,dW \tag{A.18}$$

where dW is a Wiener process. Supposing that G is function of x and t, then G follows the process:

$$dG = \left(\frac{\partial G}{\partial x}a + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial^2 x}b^2\right)dt + \frac{\partial G}{\partial x}b\,dW \tag{A.19}$$

For example, choosing $G = ln\,(x)$, we then get the following stochastic process for G (Notice that $\frac{\partial G}{\partial t} = 0$):

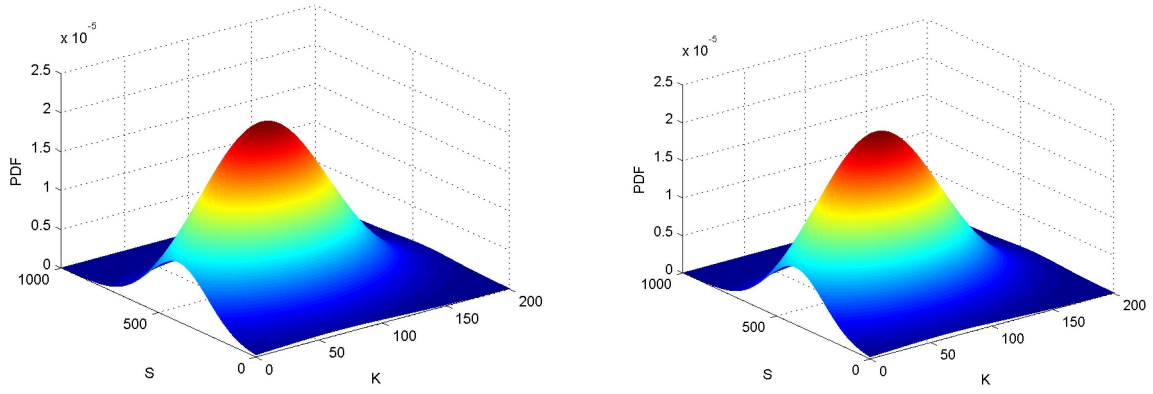
$$dG = \left(\frac{a\,(x,t)}{x} - \frac{1}{2}\frac{b\,(x,t)^2}{x^2}\right)dt + \frac{b\,(x,t)}{x}dW \tag{A.20}$$

A.6 Discrepancy Between Simulation Models In The Wild-Type

A.6.1 Discrepancy Between The 7D Gillespie And Langevin Models

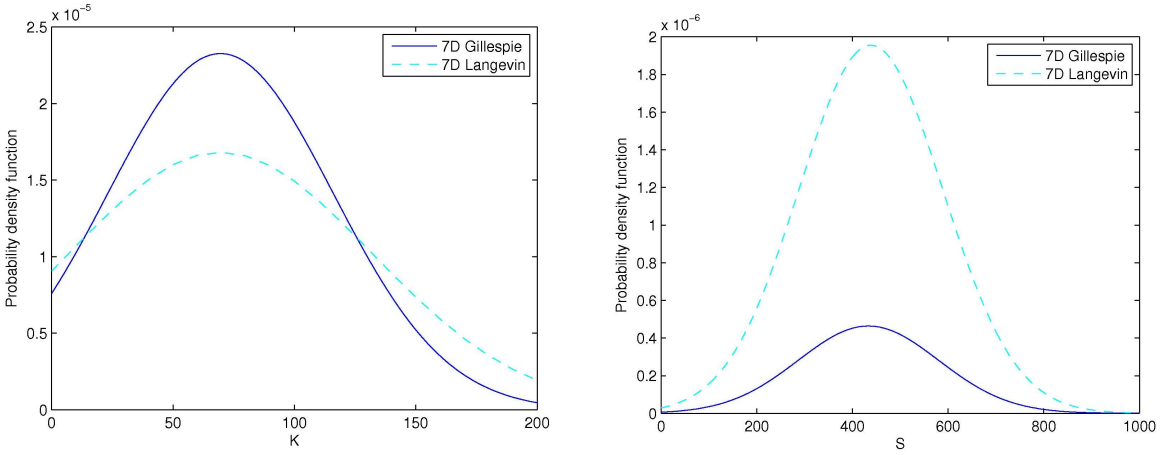
In order to compare the 7D Gillespie and Langevin models in the wild-type, we can compute the covariances for the steady state so as to come up with a probability density function (PDF) of species around the steady state, we then compare the PDFs obtained from each model. In order to compute the PDF of the 7D Gillespie model, we sample all the simulation data near the steady state (in our case, the sampled data satisfies

$0 \leq K \leq 200$, $0 \leq S \leq 1000$). We then compute the covariance matrix of the data using built-in function *cov* in Matlab, the PDF is therefore computed using function *mvnpdf*. The two probability density functions are shown in Figure A.1.



(a) (b)

FIGURE A.1: Probability density functions in the 7D Gillespie (a) and 7D Langevin models (b).



(a) (b)

FIGURE A.2: Probability density functions in the 7D Langevin and Gillespie models $f(K, S = 409)$ (a), and $f(S, K = 200)$ (b).

In order to quantify the difference between the two PDFs, we do the comparison by cutting the PDFs at the fixed point ($S^* = 409$) and projecting them on K-axis, this allows us to obtain $f(K, S = S^*)$. Similarly, we cut off the PDF at their tails ($K^* = 200$) and project them on S-axis in order to obtain $f(S, K = K^*)$. It is clear that there is a slight difference in those PDFs, especially the tail of the 7D Langevin probability density function is larger than that of the 7D Gillespie. In order to quantitatively compare the two curves shown in the figure on the left, we can compute the variances of the two data sets by applying the formula: $\text{var}(K) = \int_K (K - \mu)^2 f(K, S^*) dK$, $\mu = \int_K K f(K, S^*) dK$

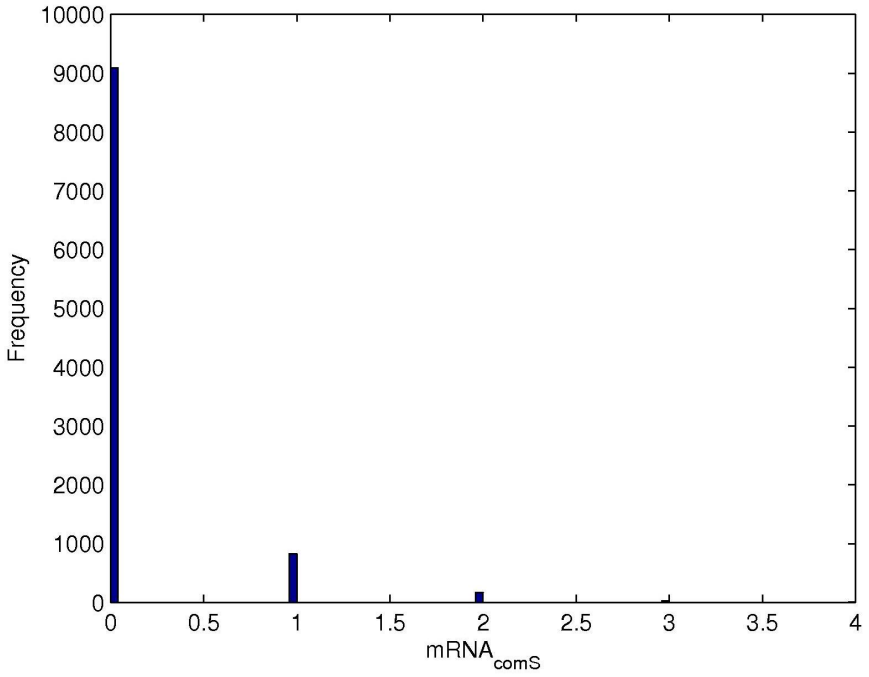


FIGURE A.3: Histogram of $mRNA_{comS}$.

where $f(K, S)$ is the joint PDF. The variances computed for the 7D Gillespie and 7D Langevin models are 17.4 and 20, respectively. This means the variance in the 7D Langevin model is about 15% larger than that in the 7D Gillespie model. This implies that the initialization probability computed in the 7D Langevin model will be larger. In order to quantify this, we can estimate the probability of ComK near the tail of the distribution where the system can possibly become competent. We suppose that for very small ΔK and particular value of S, $f(K, S)$ does not significantly change for all $200 \leq K \leq 200 + \Delta K$; therefore, $P(200 \leq K \leq 200 + \Delta K) = \int_S f(K, S) dS \Delta K$, we can take $\Delta K = 1$ for simplicity. As a result, the probability computed for the 7D Langevin model is 7.4×10^{-4} which is roughly four times as large compared to 1.75×10^{-4} in the 7D Gillespie. Consequently, the initialization probability in the 7D Langevin model is expected to be larger than that in the 7D Gillespie model. This means the Langevin approximation does not capture the right dynamics of the system. In fact, this approximation method can only work well if the populations of species are large enough. However, we found that the $mRNA_{comS}$ is present in very low number near the steady state (Figure A.3). To avoid this circumstance, we can make a change of variable to a logarithmic scale, then apply the Ito's lemma to get the corresponding Langevin equation (details of the lemma can be found in A.5). However, this solution comes at the cost of extremely slow performance since the time step needed for the simulation becomes very tiny. Consequently, it is better to eliminate this variable using the adiabatic approximation. Since the lifetime of mRNA is 50 times shorter than that of protein; therefore, the adiabatic approximation is suitable to be used in this case. In the rest of this section, I will try to reduce the 7D system to a lower-dimensional system,

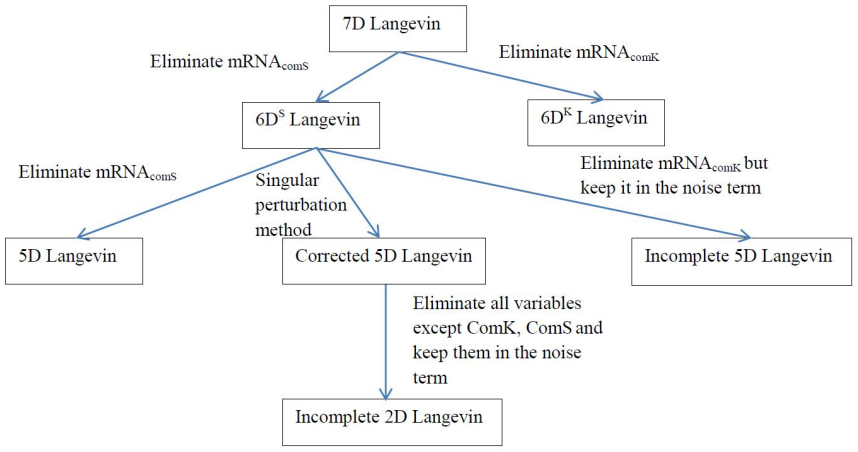


FIGURE A.4: Structure of different stochastic models. The arrows shows an approximation of high-dimensional system to a lower-dimensional system.

the structure of stochastic models we are going to work on is illustrated in Figure A.4.

A.6.2 The $6D^S$ Langevin model

This model is obtained by adiabatically eliminating $mRNA_{comS}$. We assume that the dynamics of $mRNA_{comS}$ denoted as m_s is much faster than that in the protein; therefore, we can replace this variable by its steady value:

$$m_s = \frac{k_4 + \frac{k_5 K^n}{1+(\frac{K}{k_s})^p}}{k_9} \tag{A.21}$$

The simulation result shows that the trajectories have not been trapped in the excitable state but follow the slow manifold to get back to the vegetative state (Figure A.5). However, the competent events occur more often compared to the that in the full 7D Gillespie model (Figure A.6). Figure A.7 shows a comparison in terms of PDF for the two models. By applying the same procedure as previously, we can see that the probability distribution in the $6D^S$ Langevin model (variance = 124) is much broader than that in the 7D Gillespie model (variance = 17.4). Moreover, the probability of ComK near the tail of the PDF computed for the $6D^S$ Langevin model is 7.2×10^{-4}, which is about four times larger than that in the 7D Gillespie model ($P = 1.75 \times 10^{-4}$) (see Figure A.8). Thus, we expect that the initialization probability should also be larger than that in the 7D Gillespie model. In fact, the probability for cells to become competent is computed to be about 0.11 ± 0.002 which is roughly 14 times larger than that in the wild-type ($P_{init} = 0.0076 \pm 2.3 \times 10^{-4}$). The discrepancy between P_{init} in the Gillespie and Langevin models probably comes from the fluctuation in the $mRNA_{comK}$ which may contribute a lot to the dynamics of system causing the transition to happen. In order

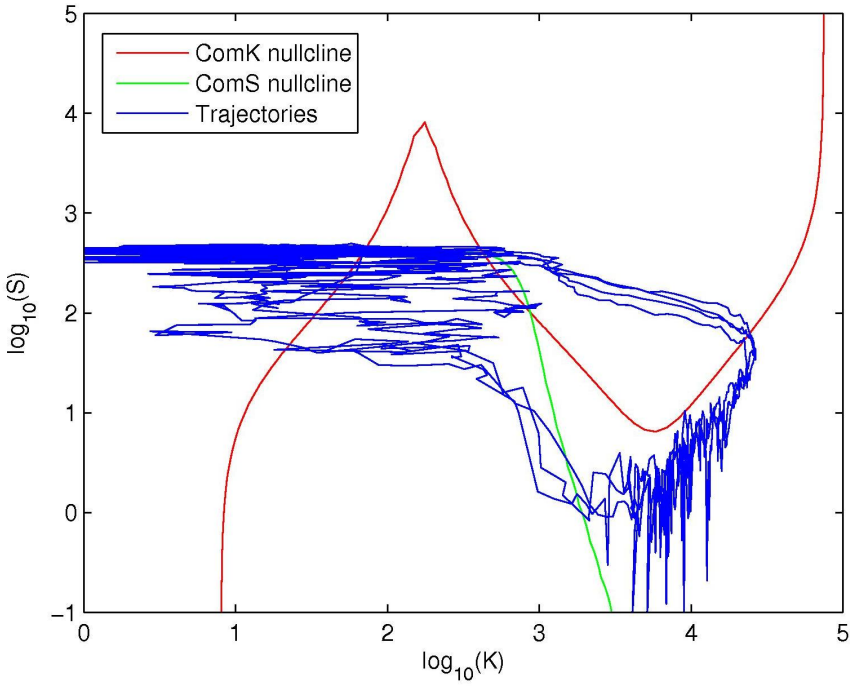
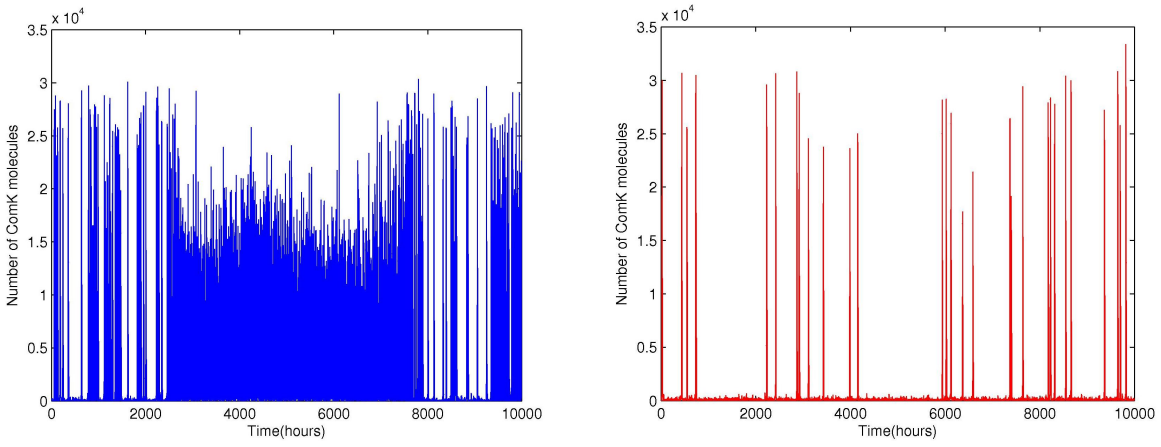


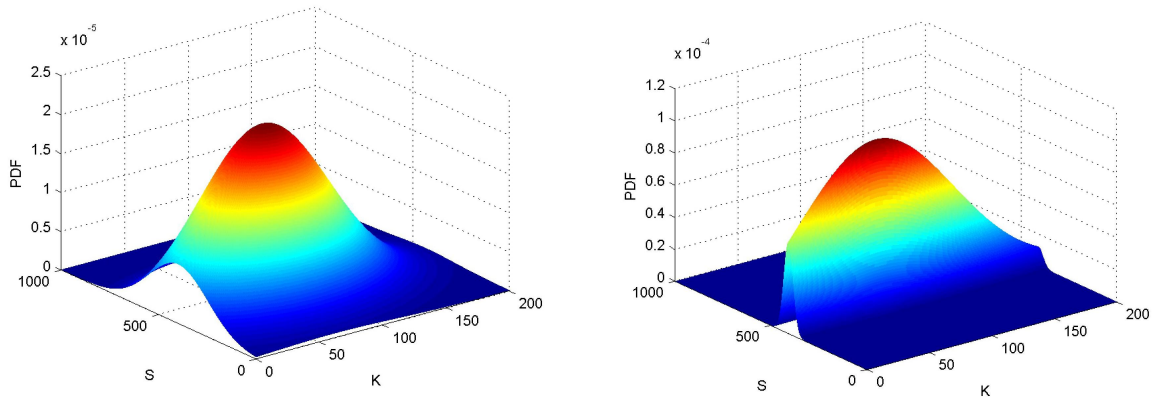
FIGURE A.5: Trajectories generated by the $6D^S$ Langevin model.



(a) (b)

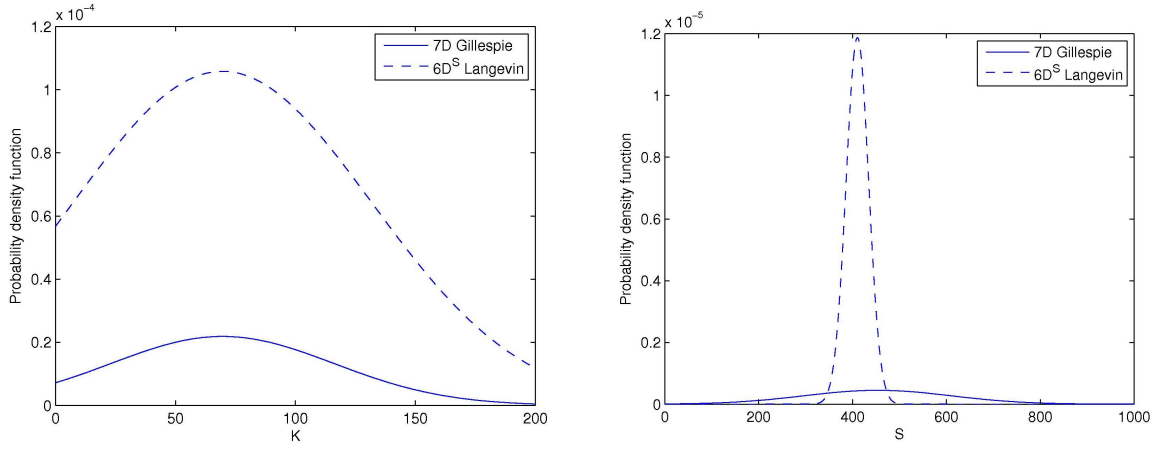
FIGURE A.6: Competent events in the $6D^S$ Langevin model (a) and the 7D Gillespie model (b).

to address this, we do the simulation while keeping the $mRNA_{comS}$ and eliminating the $mRNA_{comK}$. As a result, we then come up with a $6D^K$ Langevin model which is described in the following section.



(a) (b)

FIGURE A.7: Probability density functions in the 7D Gillespie (a) and $6D^S$ Langevin models (b).



(a) (b)

FIGURE A.8: Probability density functions in the 7D Gillespie and $6D^S$ Langevin models $f(K, S = 409)$ (a), and $f(S, K = 200)$ (b).

A.6.3 The $6D^K$ Langevin model

In this case, we eliminate $mRNA_{comK}$ denoted as m_k by setting this variable to its steady state value:

$$m_k = \frac{k_1 + \frac{k_2 K^n}{k_k^n + K^n}}{k_7} \tag{A.22}$$

As a result, we obtain a $6D^K$ Langevin model in which we find that cells will stay around the steady state and never go to the excitable state (Figure A.9). This behaviour is completely different from that in the case of eliminating $mRNA_{comS}$ and it clearly does not describe the right behaviour. Thus, we can not remove the $mRNA_{comK}$ since its dynamics plays a critical role in driving the system to the competent state. The

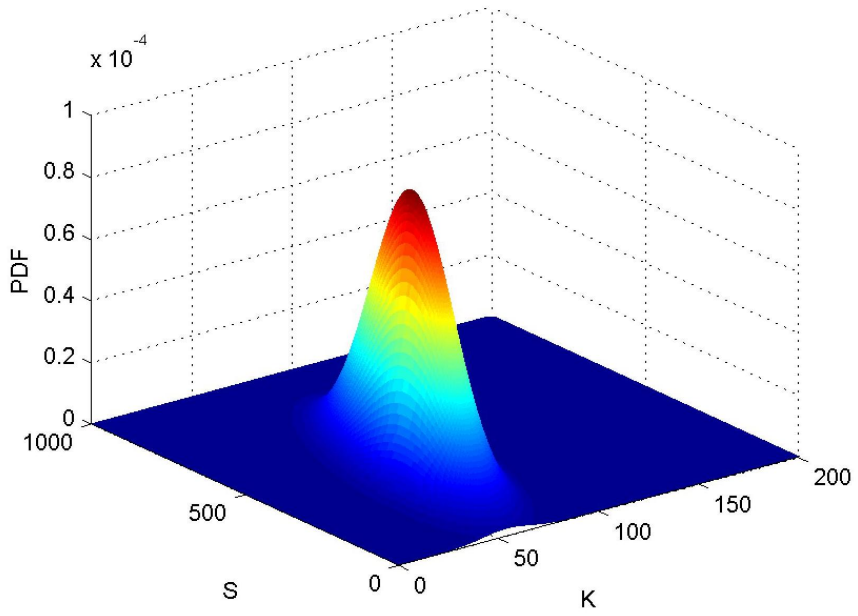


FIGURE A.9: The probability density function in the $6D^K$ Langevin model.

fluctuation in the $mRNA_{comK}$ still significantly contributes to the total noise of protein; therefore, it is necessary to put this fluctuation back into the noise term of the Langevin equation.

In the following section, we will try doing this by introducing an Incomplete 5D Langevin model in order to capture the fluctuation which has been ignored in the 6D Langevin models.

A.7 The Incomplete 5D and 2D Langevin models

In this section, we still do the model reduction from the 6D models by eliminating the $mRNA$. However, the issue is that how we can eliminate this variable but still include its fluctuation to the stochastic component in the Langevin equation for the protein. In fact, we can probably do that by treating the deterministic and stochastic terms of the Langevin equation for the variable separately. In particular, we still do the adiabatic approximation for the $mRNA_{comK}$ by assuming that the $mRNA_{comK}$ quickly reaches equilibrium, we then get:

$$m_k = \frac{k_1 + \frac{k_2 K^n}{{k_k}^n + K^n}}{k_7} \tag{A.23}$$

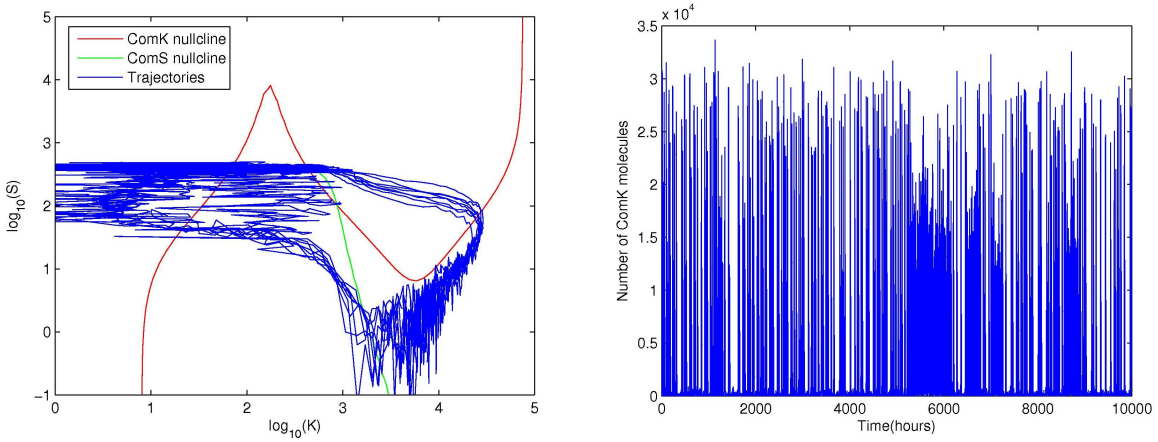
The Langevin equation for $mRNA_{comK}$ is:

$$m_k(t+dt) = m_k(t) + \left(k_1 + \frac{k_2 K^n}{{k_k}^n + K^n} - k_7 m_k(t)\right) dt + \sqrt{k_1 + \frac{k_2 K^n}{{k_k}^n + K^n} + k_7 m_k(t)}\, dW \tag{A.24}$$

From A.23 and A.24, we end up with the following new Langevin equation for $mRNA_{comK}$:

$$m_k(t+dt) = m_k(t) + \sqrt{2k_7 m_k(t)}\, dW \tag{A.25}$$

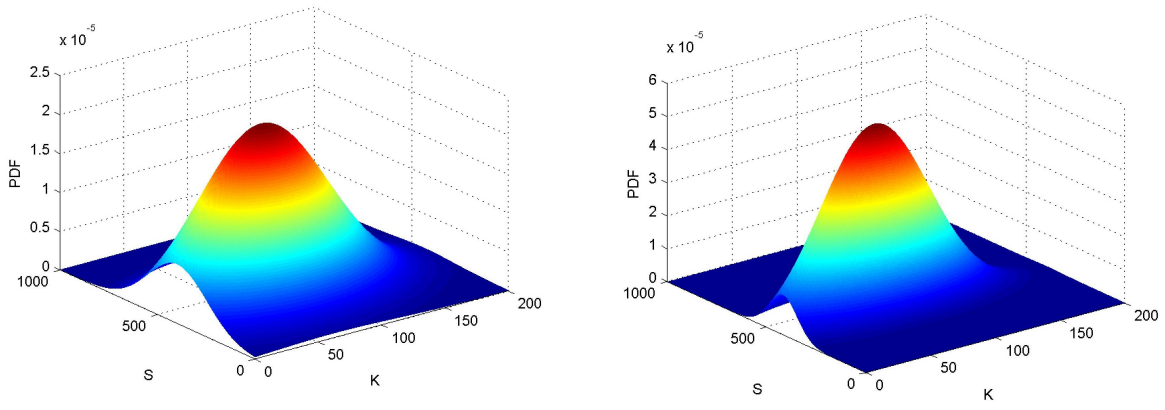
In the last equation, the $mRNA_{comK}$ evolves around its mean with the variance being expressed in Equation (A.25). As a result, we can basically eliminate this variable in the full system by doing the adiabatic approximation for the deterministic term only. However, we still keep the time-evolution of this variable and put it back into the stochastic term. Consequently, the new Langevin equation is now described by the deterministic part which only includes five variables and the stochastic part which consists of six variables. To avoid confusion, we temporarily call this new model the Incomplete 5D Langevin model. In this model, the trajectories are similar to that in the $6D^S$ Langevin model (Figure A.10).



(a) (b)

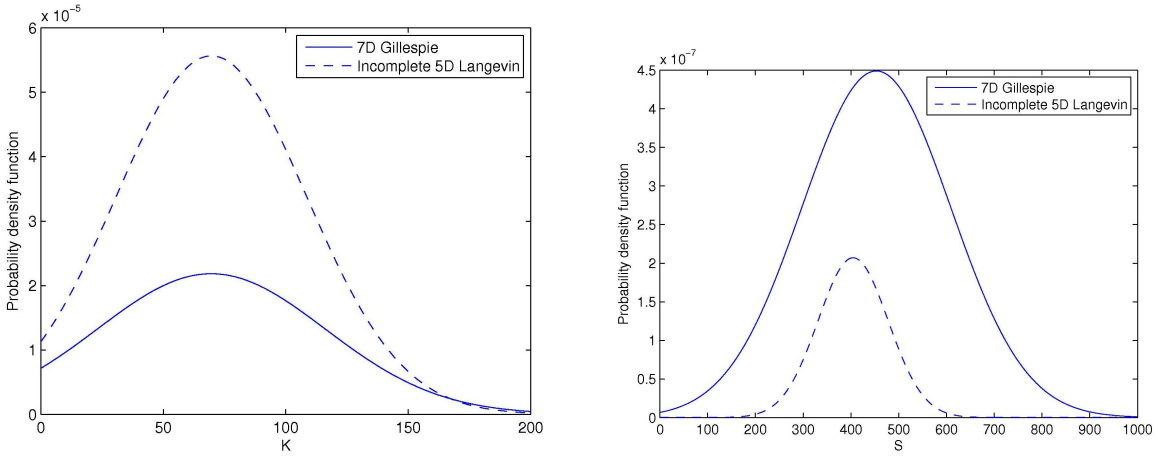
FIGURE A.10: Sample of trajectories (a), and competent events (b) in the Incomplete 5D Langevin model.

The PDF computed from the Incomplete 5D Langevin is still very different from that in the 7D Gillespie model (Figure A.11). Moreover, the initialization probability is estimated at 0.04 ± 0.007 which is smaller than that in the $6D^S$ Langevin model but still about five times as large as that in the 7D Gillespie model, though the probability of ComK in the 7D Gillespie model is 10 times larger than the Incomplete 5D Langevin model ($P = 3.8 \times 10^{-5}$). This is because the variance computed in the Incomplete 5D Langevin model is 34.3 which is roughly twice as large as that in the 7D Gillespie model (variance = 17.4), resulting in a longer tail of the PDF (Figure A.12). As a result, the probability of being competent is decreased in the Incomplete 5D Langevin model but



(a) (b)

FIGURE A.11: Probability density functions in 7D Gillespie (a) and the Incomplete 5D Langevin models (a).

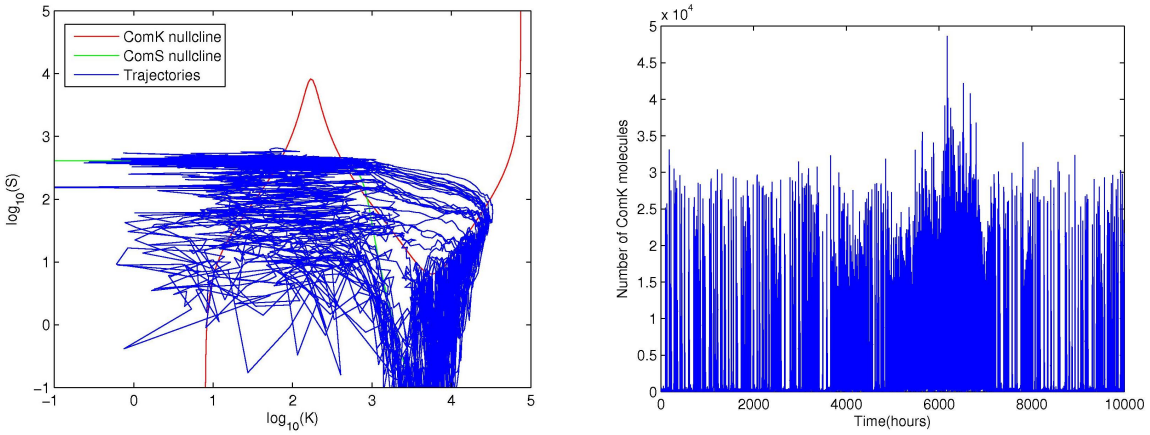


(a) (b)

FIGURE A.12: Probability density functions in 7D Gillespie and Incomplete 5D Langevin models $f(K, S = 409)$ (a), and $f(S, K = 200)$ (b).

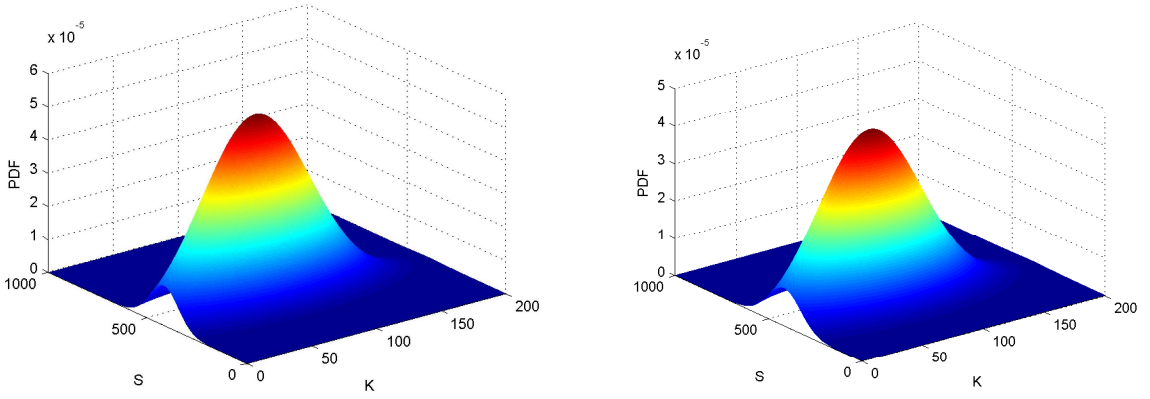
is still very high compared to that in the wild-type model. In order to verify if we end up with the same result for the lower-dimensional system, we apply the same procedure for the 2D deterministic approximate model which has been obtained in Chapter 3 so as to construct an Incomplete 2D Langevin model. In this context, we also keep the fluctuations in $MecA$, $MecA_K$ and $MecA_S$ and plug them back into the stochastic noise term of the protein.

The simulation result shows similar trajectories but the number of competent events seem to be larger compared to the Incomplete 5D Langevin model (Figure A.13) though the PDFs computed from the two models looks similar (Figure A.14). In fact, the probability of competence in the Incomplete 2D Langevin model is estimated at 0.05 ± 0.005 which is 1.2 times larger than that computed in the Incomplete 5D Langevin model



(a) (b)

FIGURE A.13: Sample of trajectories (a), and competent events (b) in the Incomplete 2D Langevin model.

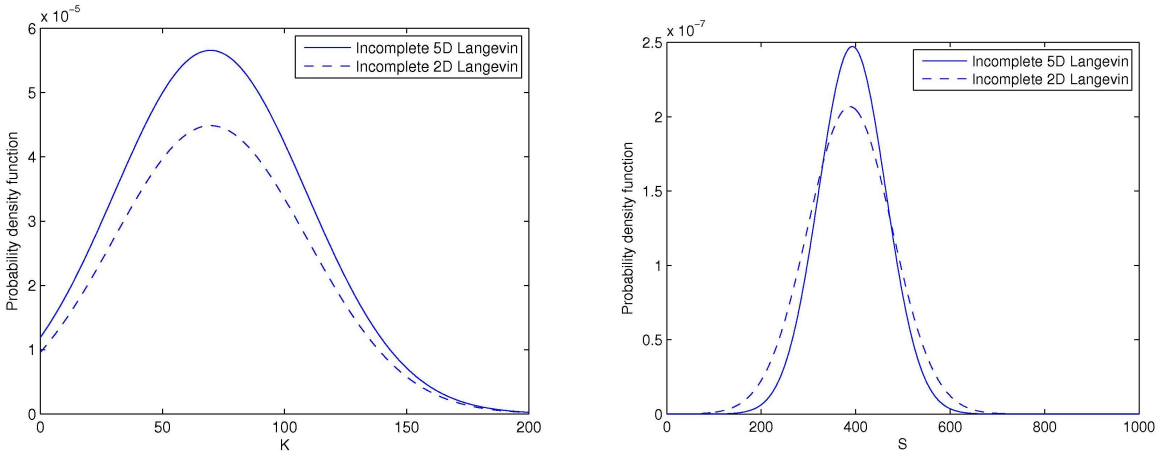


(a) (b)

FIGURE A.14: Probability density functions in the Incomplete 5D (a) and the Incomplete 2D Langevin models (b).

($P_{init} = 0.04 \pm 0.007$). Even though, the PDF near the steady state is not much different from that in the Incomplete 5D Langevin model (Figure A.15). In Figure A.15, the variance computed for the Incomplete 2D Langevin model is 28.4 which is not significantly different from that computed in the Incomplete 5D Langevin model (variance = 34.3), and the probability of ComK estimated near the tail of the distribution is 4.6×10^{-5} which is again very close to that estimated in the Incomplete 5D Langevin model ($P = 3.8 \times 10^{-5}$). Consequently, the small change near the tail of the PDF leading to significant difference in the initialization probability shows the sensitivity of those models to the tail of the distribution. This result agrees with the observation discussed in Mehta et al. (2008).

In this section, we showed our effort to approximate the dynamical behaviour of the full system by a stochastic model. However, the simulation result implied that we can not



(a) (b)

FIGURE A.15: Probability density functions in the Incomplete 2D and Incomplete 5D Langevin models $f(K, S = 409)$ (a), and $f(S, K = 200)$ (b).

do the reduction merely using the standard methods. In fact, the fluctuations from the mRNA are needed for the switching behaviour to occur. For this reason, it is critical to preserve those fluctuations in doing the model reduction. It suggests that the adiabatic approximation is not a good solution to the dimensionality reduction problem. However, by doing the reduction for the deterministic part of the Langevin equation but putting back into the system the fluctuation in the removed variables, we can reproduce a quite similar dynamical behaviour in comparison with the original system. In spite of this, one of the issue we have found when doing the reduction was the initialization probability. This quantity generally differs from that computed from the full system by order of magnitude implying that the noise has not been captured correctly. In fact, it is not obvious when doing the transformation from the reduced RRE to the reduced Langevin equation. Thus, I decided to manually construct the stochastic model by adding a tunable noise into the model. The noise terms are then altered such that we can obtain a better approximate model.

A.8 Linear Noise Approximation In The 2-Species Model

In this section, I try to identify the stochastic noise which arises from the small copy number of species as well as to evaluate the mutual effect of the mRNA on the protein noise (the details of this method can be found in section A.4).

The differential equations for the system near the fixed point (K^*, m^*) are given as follows:

$$\frac{d\langle K\rangle}{dt} = k_5 + k_3\langle m\rangle - k_6\langle K\rangle \tag{A.26}$$

$$\frac{d\langle m\rangle}{dt} = \langle f(K)\rangle - k_4\langle m\rangle \tag{A.27}$$

where $f(K) = k_1 + \frac{k_2 K^2}{{k_k}^2 + K^2}$, the Jacobian matrix $\mathbf{A}$ and diffusion matrix $\mathbf{B}$ are defined as follows:

$$\mathbf{A} = \begin{matrix} \\ (K) \\ (m) \end{matrix} \begin{matrix} \begin{matrix} (K) & (m) \end{matrix} \\ \begin{pmatrix} -k_6 & k_3 \\ \partial f(K^*)/\partial K & -k_4 \end{pmatrix} \end{matrix}, \quad \mathbf{BB}^\intercal = \begin{matrix} \\ (K) \\ (m) \end{matrix} \begin{matrix} \begin{matrix} \end{matrix} \\ \begin{pmatrix} k_5 + k_3\langle m\rangle + k_6\langle K\rangle & 0 \\ 0 & \langle f(K)\rangle + k_4\langle m\rangle \end{pmatrix} \end{matrix}$$

We define the covariance matrix $\mathbf{C}$ as follows:

$$\mathbf{C} = \begin{pmatrix} {\sigma_k}^2 & Cov(K,m) \\ Cov(K,m) & {\sigma_m}^2 \end{pmatrix}$$

Solving the Lyapunov equation for the steady state $\mathbf{AC} + \mathbf{CA}^\intercal + \mathbf{BB}^\intercal = 0$, we get:

$$-2{\sigma_k}^2 k_6 + 2k_3 Cov(K,m) + k_5 + k_3\langle m\rangle + k_6\langle K\rangle = 0 \tag{A.28}$$

$$k_3{\sigma_m}^2 + \frac{\partial f(K^*)}{\partial K}{\sigma_k}^2 - Cov(K,m)(k_4 + k_6) = 0 \tag{A.29}$$

$$\langle f(K)\rangle + k_4\langle m\rangle + \frac{2\partial f(K^*)}{\partial K} Cov(K,m) - 2k_4{\sigma_m}^2 = 0 \tag{A.30}$$

At the steady state we have:

$$k_5 + k_3\langle m\rangle - k_6\langle K\rangle = 0 \quad \Rightarrow \quad \langle m\rangle = \frac{k_6\langle K\rangle - k_5}{k_3} \tag{A.31}$$

From (A.28) and (A.31) we get:

$${\sigma_k}^2 = \langle K\rangle + \frac{k_3}{k_6} Cov(K,m) \tag{A.32}$$

On the other hand, at the steady state we have $\frac{d\langle m\rangle}{dt} \approx 0$, then $k_4\langle m\rangle = \langle f(K)\rangle$; therefore, from (A.30) we yield:

$${\sigma_m}^2 = \langle m\rangle + \frac{1}{k_4}\frac{\partial f(K^*)}{\partial K} Cov(K,m) \tag{A.33}$$

From (A.29), (A.32) and (A.33) we have:

$$\begin{aligned}(k_4 + k_6)\, Cov(K,m) &= {\sigma_m}^2 + \frac{\partial f(K^*)}{\partial K}{\sigma_k}^2 \\ &= k_3\left(\langle m\rangle + \frac{1}{k_4}\frac{\partial f(K^*)}{\partial K} Cov(K,m)\right) + \frac{\partial f(K^*)}{\partial K}\left(\langle K\rangle + \frac{k_3}{k_6} Cov(K,m)\right) \\ &= Cov(K,m)\frac{\partial f(K^*)}{\partial K}\left(\frac{k_3}{k_4} + \frac{k_3}{k_6}\right) + k_3\langle m\rangle + \langle K\rangle\frac{\partial f(K^*)}{\partial K}\end{aligned}$$

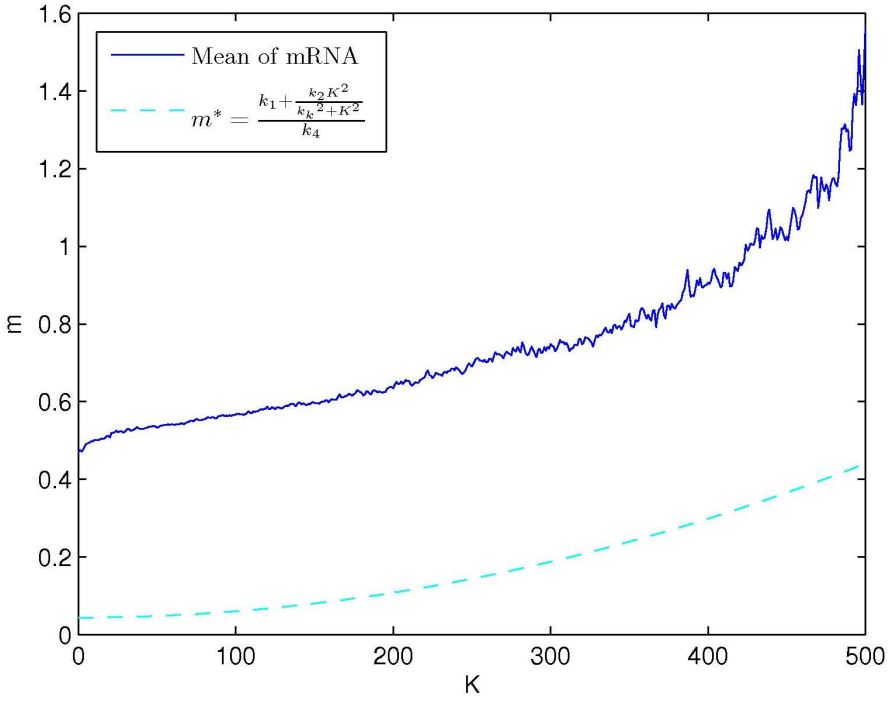


FIGURE A.16: A comparison between the mean of mRNA given by (4.18) (Chapter 4) and its steady state value.

Therefore,

$$Cov(K, m) = \frac{k_3 \langle m \rangle + \frac{\partial f(K^*)}{\partial K} \langle K \rangle}{(k_6 + k_4)\left(1 - \frac{k_3 \frac{\partial f(K^*)}{\partial K}}{k_6 k_4}\right)} \tag{A.34}$$

Plugging (A.34) into (A.32), we obtain

$${\sigma_k}^2 = \langle K \rangle + \frac{k_3}{k_6} \frac{k_3 \langle m \rangle + \frac{\partial f(K^*)}{\partial K} \langle K \rangle}{(k_6 + k_4)\left(1 - \frac{k_3 \frac{\partial f(K^*)}{\partial K}}{k_6 k_4}\right)} \tag{A.35}$$

Next, we need to compare the mean of mRNA computed with that obtained from the corresponding reaction rate equations, this will give us a clue of why the adiabatic approximation produced such a poor model as described previously.

Figure A.16 shows this comparison in which the big gap between the two curves implies that the adiabatic approximation is not good solution to the reduction problem. In fact, the means of mRNA computed from both cases will match when the propensity functions are linear; however, since we have non-linear propensity function $a = k_1 + \frac{k_2 K^2}{k_k{}^2 + K^2}$, this is not always true. By using curve fitting solver in Matlab, we can fit the mean of mRNA by a quadratic curve $(e_0 K^2 + e_1 K + e_2)$ where $e_0 = 3.4 \times 10^{-6}$, $e_1 = -3.3 \times 10^{-4}$, $e_2 = 0.55$ (Figure A.17). As a result, we can reduce our model to a 1D model by replacing $m = e_0 K^2 + e_1 K + e_2$, the stochastic process of ComK now becomes:

$$dK = g(K)dt + \eta_k \, dW_k \tag{A.36}$$

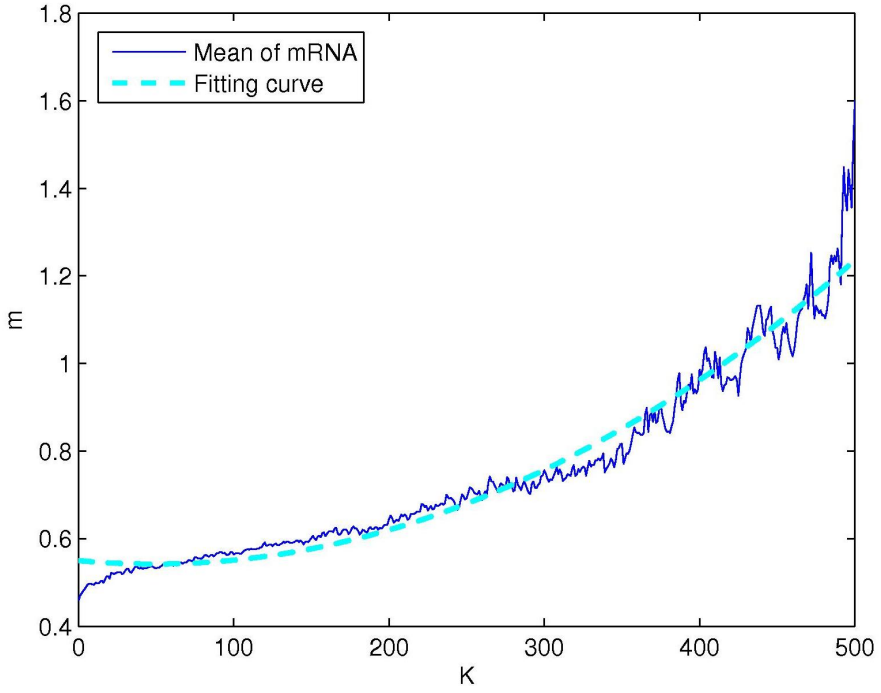


FIGURE A.17: A fitting curve of the mean of mRNA.

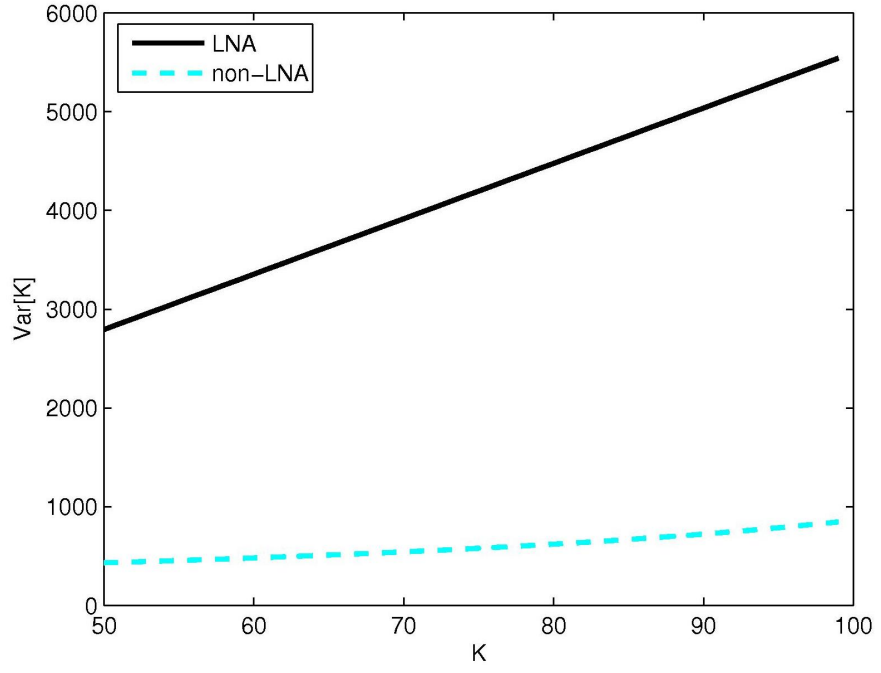


FIGURE A.18: A comparison between the variance of ComK computed from the linear noise approximation (LNA) and that computed from the empirical data (non-LNA).

where $\eta_k = e_0K^2+e_1K+e_2$, $g(K) = \big(k_5 + k_3(e_0K^2 + e_1K + e_2) - k_6K\big)$. We now apply the linear noise approximation for this reduced model, the Jacobian matrix $\mathbf{A}$, diffusion matrix $\mathbf{B}$ and covariance matrix $\mathbf{C}$ now become:

$$\mathbf{A} = \begin{matrix} \\ (K) \\ (m) \end{matrix} \begin{matrix} \begin{matrix} (K) & (m) \end{matrix} \\ \begin{pmatrix} \partial g(K^*)/\partial K & 0 \\ 0 & 0 \end{pmatrix} \end{matrix}, \quad \mathbf{BB}^\intercal = \begin{matrix} \\ (K) \\ (m) \end{matrix} \begin{matrix} \\ \begin{pmatrix} (b_0K^2 + b_1K + b_2) & 0 \\ 0 & 0 \end{pmatrix} \end{matrix}$$

$$\mathbf{C} = \begin{pmatrix} \widetilde{\sigma_k}^2 & 0 \\ 0 & 0 \end{pmatrix}$$

Solving the Lyapunov equation for the steady state $\mathbf{AC} + \mathbf{CA}^\intercal + \mathbf{BB}^\intercal = 0$, we get:

$$2\frac{\partial g(K^*)}{\partial K}\widetilde{\sigma_k}^2 + (b_0K^2 + b_1K + b_2) = 0 \tag{A.37}$$

Therefore,

$$\begin{aligned} \widetilde{\sigma_k}^2 &= -\frac{(b_0K^2 + b_1K + b_2)}{2\frac{\partial g(K^*)}{\partial K}} \\ &= -\frac{(b_0K^2 + b_1K + b_2)}{2(2k_3e_0K + k_3e_1 - k_6)} \end{aligned} \tag{A.38}$$

We now compare the variance of ComK defined by (A.35) using the linear noise approximation (LNP) and that defined by (A.38) using the empirical data (non-LNA). If the linear noise approximation can capture the noise correctly, then the variance of ComK should be the same as that computed from the reduced model. However, there is a big gap between the two quantities (Figure A.18). In fact, those quantities differ from each other by a factor of 4. This result suggests that the linear noise approximation may not well measure the local fluctuation near the stable fixed point in a bistable model, where any perturbation away from the fixed point can well be pushed to the other fixed point.

The empirical fluctuation gives us a clue of how the noise looks like, but it does not tell us how much noise we need to put back into the reduced model in order to reproduce the dynamical behaviour of the system. This is because the mean of mRNA has not been captured correctly as expected. This is the key issue that has prevented us from getting the correct fluctuation for the reduced model using adiabatic approximation. In fact, we can fix the mean of mRNA by using the fitting curve $b_0K^2 + b_1K + b_2$, this allows us to obtain the following ODE:

$$\begin{aligned} \frac{dK}{dt} &= k_5 + k_3m - k_6K \\ \frac{dm}{dt} &= k_4(b_0K^2 + b_1K + b_2) - k_4m \end{aligned} \tag{A.39}$$

However, this ODE does not have the same fixed points as that in the original model (Chapter 4, equation (4.2)). Indeed, let (K^*, m^*) is the fixed point of the model (4.2).

From the data shown on Figure A.16, we have

$$\langle m\rangle \approx b_0K^2 + b_1K + b_2 > m^* = \frac{k_1 + \frac{k_2K^2}{k_k^2+K^2}}{k_4} \quad \text{for all} \quad 0 \leq K \leq 500$$

therefore,

$$b_0K^{*2} + b_1K^* + b_2 > m^* \Rightarrow k_4(b_0K^{*2} + b_1K^* + b_2) - k_4m^* > 0$$

This means (K^*, m^*) is not the fixed point of the model (A.39). For this reason, we still need to use the adiabatic approximation in order to preserve the structure of fixed points and fit the fluctuation in the stochastic reduced model by using fitting curves, this allows us to construct a tunable noise for the stochastic model which can be used to produce a good approximation to the original 2D model.

A.9 A Finite Difference Method for The Fokker-Planck Equation

In this section, we are going to introduce an explicit method called *finite difference* for solving the Fokker-Planck equation given by $\mathbf{A}\widetilde{\mathbf{P}}^n = 0$ in Chapter 5. In order to build up matrix $\mathbf{A}$, we need to write matrix $\widetilde{\mathbf{P}}$ in form of vector. Let us do a mapping between a two-dimensional point at (i, j) of matrix $\widetilde{\mathbf{P}}^n_{i,j}$ to an one-dimensional point in its vector form by defining:

$$\widetilde{\mathbf{P}}(k) = \widetilde{\mathbf{P}}^n_{i,j}, \quad k = M(i-1)+j, \quad i = 1, 2, \ldots, M \quad j = 1, 2, \ldots, M \tag{A.40}$$

Therefore, matrix $\mathbf{A}$ can be then built up as follows:

$$\begin{aligned}
\mathbf{A}(M(i-1)+j, Mi+j) &= -\frac{a^n_{i+1,j}}{2\Delta_x} + \frac{\alpha^n_{i+1,j}}{2\Delta_x^2}, \quad i = 2, 3, \ldots, M-1 \quad j = 2, 3, \ldots, M-1 \\
\mathbf{A}(M(i-1)+j, M(i-1)+j+1) &= -\frac{b^n_{i,j+1}}{2\Delta_y} + \frac{\beta^n_{i,j+1}}{2\Delta_y^2}, \quad i = 2, 3, \ldots, M-1 \quad j = 2, 3, \ldots, M-1 \\
\mathbf{A}(M(i-1)+j, M(i-1)+j-1) &= \frac{b^n_{i,j-1}}{2\Delta_y} + \frac{\beta^n_{i,j-1}}{2\Delta_y^2}, \quad i = 2, 3, \ldots, M-1 \quad j = 2, 3, \ldots, M-1 \\
\mathbf{A}(M(i-1)+j, M(i-2)+j) &= \frac{a^n_{i-1,j}}{2\Delta_x} + \frac{\alpha^n_{i-1,j}}{2\Delta_x^2}, \quad i = 2, 3, \ldots, M-1 \quad j = 2, 3, \ldots, M-1 \\
\mathbf{A}(M(i-1)+j, M(i-1)+j) &= -(\frac{\alpha^n_{i,j}}{\Delta_x^2} + \frac{\beta^n_{i,j}}{\Delta_y^2}), \quad i = 2, 3, \ldots, M \quad j = 2, 3, \ldots, M-1
\end{aligned} \tag{A.41}$$

In fact, in order to compute the second derivatives at a particular grid point (central point), it is required of the data from its four neighbors namely top (t), bottom (b), right (r) and left (l) (see Figure A.19). As a result, from (A.41), we define the corresponding

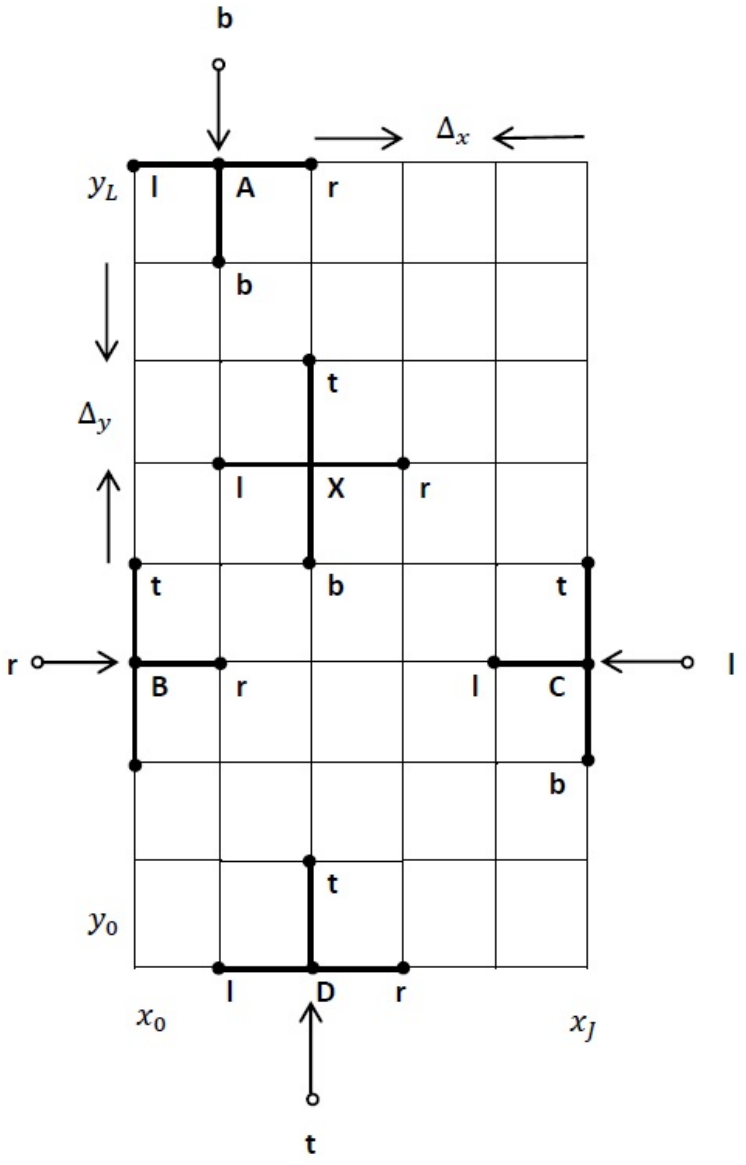


FIGURE A.19: Finite-difference representation on a two-dimensional grid. The second derivative at the point X is evaluated using the points to which A is shown connected. The second derivatives at points A,B,C,D are computed using the connected points and also using "boundary points" shown as empty circles.

terms which are evaluated at the right, top, bottom, left and central points as follows:

$$
\begin{aligned}
q_r(i+1,j) &= -\frac{a^n_{i+1,j}}{2\Delta_x} + \frac{\alpha^n_{i+1,j}}{2\Delta_x^2}, \quad i = 2,3,\ldots,M-1 \quad j = 2,3,\ldots,M-1 \\
q_t(i,j+1) &= -\frac{b^n_{i,j+1}}{2\Delta_y} + \frac{\beta^n_{i,j+1}}{2\Delta_y^2}, \quad i = 2,3,\ldots,M-1 \quad j = 2,3,\ldots,M-1 \\
q_b(i,j-1) &= \frac{b^n_{i,j-1}}{2\Delta_y} + \frac{\beta^n_{i,j-1}}{2\Delta_y^2}, \quad i = 2,3,\ldots,M-1 \quad j = 2,3,\ldots,M-1 \\
q_l(i-1,j) &= \frac{a^n_{i-1,j}}{2\Delta_x} + \frac{\alpha^n_{i-1,j}}{2\Delta_x^2}, \quad i = 2,3,\ldots,M-1 \quad j = 2,3,\ldots,M-1 \\
q_c(i,j) &= -(\frac{\alpha^n_{i,j}}{\Delta_x^2} + \frac{\beta^n_{i,j}}{\Delta_y^2}), \quad i = 2,3,\ldots,M \quad j = 2,3,\ldots,M-1
\end{aligned}
\tag{A.42}
$$

The points where $i = M, i = 0, j = M, j = 0$ are boundary points where their derivatives can not be identified since these points are outside the grid point. Hence, the boundary condition is applied in order to make sure that there is no flux at boundary. As a result, the value of central point evaluated at the boundary will be added up to the boundary information which is located at the mirror image to the original source as shown on Figure A.19. For example, the values of elements evaluated at the boundary points A,B will be added up by q_b and q_r, respectively. In general, we have the following assigning scheme for the boundary condition:

$$\begin{aligned}
A(M(M-1)+i, M(M-1)+i) &:= A(M(M-1)+i, M(M-1)+i) + q_b(M,i) \quad i = 1, 2, \ldots, M \\
A(i,i) &:= A(i,i) + q_t(i,i) \quad i = 1, 2, \ldots, M \\
A(i(i-1)+M, i(i-1)+M) &:= A(i(i-1)+M, i(i-1)+M) + q_l(i,M) \quad i = 1, 2, \ldots, M \\
A(i(i-1)+1, i(i-1)+1) &:= A(i(i-1)+1, i(i-1)+1) + q_r(i,1) \quad i = 1, 2, \ldots, M
\end{aligned} \tag{A.43}$$

We now have to find the solution to the equation $\mathbf{A}\widetilde{\mathbf{P}} = 0$. In fact, this is equivalent with finding the eigenvector of $\mathbf{A}$ with respect to zero eigenvalue. However, since A is a very large sparse matrix, it may require a lot of memory to store the data. In addition, matrix $\mathbf{A}$ is not always positive definite unless $\Delta_x, \Delta_y \ll 1$. This condition again may slow down the performance since it requires more data points for the calculation. To overcome this, we first start with a small number of data points, then gradually reduce Δ_x or Δ_y such as the condition is satisfied.

A.10 Langevin Simulation

In order to generate trajectories using Langevin approximation, we first need to construct the Langevin equation for the system. The set of chemical reactions r_i $(i = 1, 2, \ldots 16)$ of the full system $\mathbf{X} = (K, S, A, A_K, A_S, R_K, R_S)^\intercal$ can be re-written as follows (we use

the same symbols for variables as mentioned in the main text):

$$
\begin{aligned}
r_1,&\quad \mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} \xrightarrow{k_1} \mathrm{P}_{\mathrm{comK}}^{\mathrm{const}} + \mathrm{R_K}\\
r_2,&\quad \mathrm{P_{comK}} \xrightarrow{f(K,k_2,k_k,n)} \mathrm{P_{comK}} + \mathrm{R_K}\\
r_3,&\quad \mathrm{R_K} \xrightarrow{k_3} \mathrm{R_K} + \mathrm{K}\\
r_4,&\quad \mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} \xrightarrow{k_4} \mathrm{P}_{\mathrm{comS}}^{\mathrm{const}} + \mathrm{R_S}\\
r_5,&\quad \mathrm{P_{comS}} \xrightarrow{g(K,k_5,k_s,p)} \mathrm{P_{comS}} + \mathrm{R_S}\\
r_6,&\quad \mathrm{R_S} \xrightarrow{k_6} \mathrm{R_S} + \mathrm{S}\\
r_7,&\quad \mathrm{R_K} \xrightarrow{k_7}\\
r_8,&\quad \mathrm{K} \xrightarrow{k_8} \emptyset\\
r_9,&\quad \mathrm{R_S} \xrightarrow{k_9}\\
r_{10},&\quad \mathrm{S} \xrightarrow{k_{10}} \emptyset\\
r_{11},&\quad \mathrm{A} + \mathrm{K} \xrightarrow{k_{11}/\Omega} \mathrm{A_K}\\
r_{12},&\quad \mathrm{A_K} \xrightarrow{k_{-11}} \mathrm{A} + \mathrm{K}\\
r_{13},&\quad \mathrm{A_K} \xrightarrow{k_{12}} \mathrm{A}\\
r_{14},&\quad \mathrm{A} + \mathrm{S} \xrightarrow{k_{13}} \mathrm{A_S}\\
r_{15},&\quad \mathrm{A_S} \xrightarrow{k_{-13}/\Omega} \mathrm{A} + \mathrm{S}\\
r_{16},&\quad \mathrm{A_S} \xrightarrow{k_{14}} \mathrm{A}
\end{aligned}
\tag{A.44}
$$

The propensity functions a_i $(i = 1, 2, \ldots 16)$ are detailed below:

$$
\begin{aligned}
a_1 &= k_1\\
a_2 &= \frac{k_2 K^n}{{k_k}^n + K^n}\\
a_3 &= k_3 R_K\\
a_4 &= k_4\\
a_5 &= \frac{k_5}{1 + (K/k_s)^p}\\
a_6 &= k_6 R_S\\
a_7 &= k_7 R_K\\
a_8 &= k_8 K\\
a_9 &= k_9 R_S\\
a_{10} &= k_{10} S\\
a_{11} &= k_{11} K A\\
a_{12} &= k_{-11} M_K
\end{aligned}
$$

$$a_{13} = k_{13}SA$$
$$a_{14} = k_{-13}M_S$$
$$a_{15} = k_{12}M_K$$
$$a_{16} = k_{14}M_S$$

From (A.44), the stoichiometry vectors $\boldsymbol{\nu}_j$, $j = 1, 2, \ldots 16$ which represent the change in the amount of molecular numbers in $\mathbf{X}$ are defined as follows:

$$\boldsymbol{\nu}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \boldsymbol{\nu}_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \boldsymbol{\nu}_6 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix},$$

$$\boldsymbol{\nu}_8 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_9 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \boldsymbol{\nu}_{10} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_{11} = \begin{pmatrix} -1 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_{12} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_{13} = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$\boldsymbol{\nu}_{14} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_{15} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\nu}_{16} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

Applying the formula of the Langevin equation mentioned in section 1.1.3 of Chapter 1,

we end up with the following Langevin equation for the system:

$$\begin{aligned}
K(t+\Delta t) &= K(t) + (a_{12} - a_{11} + a_3 - a_8)\,\Delta t + (-W_{11}\sqrt{a_{11}} + W_{12}\sqrt{a_{12}} + W_3\sqrt{a_3} - W_8\sqrt{a_8})\,\sqrt{\Delta t} \\
S(t+\Delta t) &= S(t) + (a_{14} - a_{13} - a_{10} + a_6)\,\Delta t + (-W_{10}\sqrt{a_{10}} - W_{13}\sqrt{a_{13}} + W_{14}\sqrt{a_{14}} - W_6\sqrt{a_6})\,\sqrt{\Delta t} \\
A(t+\Delta t) &= A(t) + (a_{12} - a_{11} - a_{13} + a_{14} + a_{15} + a_{16})\,\Delta t \\
&\quad + (-W_{11}\sqrt{a_{11}} + W_{12}\sqrt{a_{12}} - W_{13}\sqrt{a_{13}} + W_{14}\sqrt{a_{14}} + W_{15}\sqrt{a_{15}} + W_{16}\sqrt{a_{16}})\,\sqrt{\Delta t} \\
M_K(t+\Delta t) &= M_K(t) + (a_{11} - a_{12} - a_{15})\,\Delta t + (W_{11}\sqrt{a_{11}} - W_{12}\sqrt{a_{12}} - W_{15}\sqrt{a_{15}})\,\sqrt{\Delta t} \\
M_S(t+\Delta t) &= M_S(t) + (a_{13} - a_{14} - a_{16})\,\Delta t + (W_{13}\sqrt{a_{13}} - W_{14}\sqrt{a_{14}} - W_{16}\sqrt{a_{16}})\,\sqrt{\Delta t} \\
R_K(t+\Delta t) &= R_K(t) + (a_1 + a_2 - a_7)\,\Delta t + (W_1\sqrt{a_1} + W_2\sqrt{a_2} - W_7\sqrt{a_7})\,\sqrt{\Delta t} \\
R_S(t+\Delta t) &= R_S(t) + (a_4 + a_5 - a_9)\,\Delta t + (W_4\sqrt{a_4} + W_5\sqrt{a_5} - W_9\sqrt{a_9})\,\sqrt{\Delta t}
\end{aligned} \tag{A.45}$$

where $W_i = \mathcal{N}_i(0,1)$, $i = 1, 2, \ldots 16$. In simulation, W_i are Gaussian distributed random variables with mean 0, variance 1; therefore, they can be easily generated using simple algorithm in C++ or Matlab. The updating scheme given by (A.45) allows us to generate trajectories of the system. In principle, the algorithm can be invalid if there is any variable driven to negative values. We can avoid this by checking if the variables are valid for updating, if not then set them to their previous values, reducing the time step by half and start again. This procedure can be iterated until the duration of the simulation is up.

A.11 Dizzy Simulation

The simulation data used in the main text is generated by running the Gillespie simulation using Dizzy, the Dizzy file is detailed below:

$$\begin{aligned}
Pconst_{comK} &= 1; \\
mRNA_{comK} &= 0; \\
P_{comK} &= 1; \\
ComK &= 69; \\
ComS &= 409; \\
Pconst_{comS} &= 1; \\
P_{comS} &= 1; \\
mRNA_{comS} &= 0; \\
MecA_K &= 0; \\
MecA_S &= 477; \\
MecA &= 23;
\end{aligned}$$

$$
\begin{aligned}
k_1 &= 0.00021875;\\
k_2 &= 0.1875;\\
k_3 &= 0.2;\\
k_4 &= 0;\\
k_5 &= 0.0015;\\
k_6 &= 0.2;\\
k_7 &= 0.005;\\
k_8 &= 1e-4;\\
k_9 &= 0.005;\\
k_{10} &= 1e-4;\\
k_{11} &= 2.02e-6;\\
k_{-11} &= 5e-4;\\
k_{12} &= 0.05;\\
k_{13} &= 4.5e-6;\\
k_{-13} &= 5e-5;\\
k_{14} &= 4e-5;\\
k_k &= 5000;\\
k_s &= 833;\\
n &= 2;\\
p &= 5;
\end{aligned}
$$

$r1,\quad Pconst_{comK} \rightarrow Pconst_{comK} + mRNA_{comK}, k_1;$

$r2,\quad P_{comK} \rightarrow P_{comK} + mRNA_{comK}, [k_2 * ComK^n/(k_k^n + ComK^n)];$

$r3,\quad mRNA_{comK} \rightarrow mRNA_{comK} + ComK, k_3;$

$r4,\quad Pconst_{comS} \rightarrow Pconst_{comS} + mRNA_{comS}, k_4;$

$r5,\quad P_{comS} \rightarrow P_{comS} + mRNA_{comS}, [k_5/(1 + (ComK/k_s)^p)];$

$r6,\quad mRNA_{comS} \rightarrow mRNA_{comS} + ComS, k_6;$

$r7,\quad mRNA_{comK} \rightarrow, k_7;$

$r8,\quad ComK \rightarrow, k_8;$

$r9,\quad mRNA_{comS} \rightarrow, k_9;$

$r10,\quad ComS \rightarrow, k_{10};$

$r11,\quad MecA + ComK \rightarrow MecA_K, k_{11};$

$r12,\quad MecA_K \rightarrow MecA + ComK, k_{-11};$

$r13,\quad MecA + ComS \rightarrow MecA_S, k_{13};$

$r14, \quad MecA_S \rightarrow MecA + ComS, k_{-13};$

$r15, \quad MecA_K \rightarrow MecA, k_{12};$

$r16, \quad MecA_S \rightarrow MecA, k_{14};$

Bibliography

Alberts, B. (2002). *Molecular biology of the cell*. New York : Garland Science, 4th ed. edition.

Andrey, R., Chenyu, L., Richard, H., and Terence, R. (2006). Abortive initiation and productive initiation by rna. *Science*, 314(5802):1139–1143.

Avi, M. and Ben, D. M., editors (2012). *New Frontiers of network Analysis in Systems Biology*. Springer New York.

Bennett, M., Volfson, D., Tsimring, L., and Hasty, J. (2007). Transient dynamics of genetic regulatory networks. *Biophysical journal*, 92(10):3501–12.

Berglund, N. and Gentz, B. (2003). Geometric singular perturbation theory for stochastic differential equations. *Journal of Differential Equations*, 191(1):1–54.

Bowman, A. (1984). An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353.

Cagatay, T., Turcotte, M., Elowitz, M., Garcia-Ojalvo, J., and Süel, G. (2009a). Architecture-dependent noise discriminates functionally analogous differentiation circuits. *Cell*, 139(3):512–22.

Cagatay, T., Turcotte, M., Elowitz, M., Garcia-Ojalvo, J., and Süel, G. (2009b). Supplemental data architecture-dependent noise discriminates functionally analogous differentiation circuits. *Cell*, 139(3):1–29.

Callis, J. (1995). Regulation of protein degradation. *The Plant Cell*, 7(7):845–857.

Carey, Michael, S., and Stephen, T. (1999). *Transcriptional Regulation in Eukaryotes : Concepts, Strategies, & Techniques*. Cold Spring Harbor Laboratory Press.

Cazzaniga, P., Pescini, D., Besozzi, D., and Mauri, G. (2006). Tau leaping stochastic simulation method in p systems. *Computer*, pages 298–313.

Cheemeng, T., Philippe, M., and Lingchong, Y. (2009). Emergent bistability by a growth-modulating positive feedback circuit. *Nature Chemical Biology*, 5(11):842–848.

Cotter, S., Zygalakis, K., Kevrekidis, I., and Erban, R. (2011). A constained approach to multiscale stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 135(9):094120.

Dandach, S. and Khammash, M. (2010). Analysis of stochastic strategies in bacterial competence: A master equation approach. *PLoS Computational Biology*, 6(11):e1000985.

Day, D. and Tuite, M. F. (1998). Post-transcriptional gene regulatory mechanisms in eukaryotes: an overview. *The Journal of endocrinology*, 157(3):361–71.

Dehling, Gottschalk, H. G., Hoffmann, T., and Alex, C. (2007). *Stochastic Modelling in Process Technology*. Elsevier Science & Technology.

Drew, J., Glen, A., and Leemis, L. (2000). Computing the cumulative distribution function of the kolmogorovsmirnov statistic. *Computational Statistics & Data Analysis*, 34(1):1–15.

Dubnau, D. (1991). Genetic competence in bacillus subtilis. *Microbiology and Molecular Biology Reviews*, 55(3):395–424.

Dubnau, D. and Losick, R. (2006). Bistability in bacteria. *Molecular microbiology*, 61(3):564–72.

Elowitz, M., Levine, A., Siggia, E., and Swain, P. (2002). Stochastic gene expression in a single cell. *Science (New York, N.Y.)*, 297(5584):1183–6.

Facoltá, F., Matematiche, F., Naturali, F., Gilmore, S., and Cacchiani, L. (2007). *Simulation and Analysis of Chemical Reactions using Stochastic Differential Equations*. PhD thesis, Università Degli Studi Di Torino.

Fraser, S. (1988). The steady state and equilibrium approximations: A geometrical picture. *The Journal of Chemical Physics*, 88(8):4732.

Fuglede, B. and Topsoe, F. (2004). Jensen-shannon divergence and hilbert space embedding. *International Symposium on Information Theory*, pages 31–31.

Gardner, T., Cantor, C., and Collins, J. (2000). Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–42.

Geoffrey, M. and Cooper (2000). *The Cell: A Molecular Approach*. Sunderland (MA) : Sinauer Associates, 2nd edition.

Gillespie, D. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

Gillespie, D. (2000). The chemical langevin equation. *Journal of Chemical Physics*, 113(1):297–306.

Gillespie, D. (2002). The chemical langevin and fokker-planck equations for the reversible isomerization reactions. *The Journal of Physical Chemistry A*, 106:5063–5071.

Gillespie, D. (2007). Stochastic simulation of chemical kinetics. *Annual review of physical chemistry*, 58:35–55.

Hatami, N., Naderkhani, G., Shah, N., and Young, B. (2004). The effects of pretreatment of competent cells with nalidixic acid on efficiency of chemically-induced transformation in escherichia coli b23. *Journal of Experimental Microbiology and Immunology (JEMI)*, 12:84–87.

Herzenberg, L., Parks, D., Sahaf, B., Perez, O., and Roederer, M. (2002). The history and future of the fluorescence activated cell sorter and flow cytometry: a view from stanford. *Clinical chemistry*, 48(10):1819–27.

Hill, A. V. (1910). The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *The journal of physiology*, 40:iv–vii.

Hill, A. V. (1913). The combinations of haemoglobin with oxygen and with carbon monoxide. i. *Biochemical Journal*, 7(5):471–80.

Hinchcliffe, E. (2005). Essay using long-term time-lapse imaging of mammalian cell cycle progression for laboratory instruction and analysis. *Cell Biology Education*, 4:284–290.

Holmes, M. (2009). *Introduction to the Foundations of Applied Mathematics*. Springer New York.

Johnson, D. and Sinanovic, S. (2001). Symmetrizing the kullback-leibler distance. *IEEE Transactions on Information Theory*.

Justel, A., Pefia, D., and Zamar, R. (1997). A multivariate kolmogorov-smimov test of goodness of fit 1. *Statistics & Probability Letters*, 2:251–259.

Kitano, H. (2002). Systems biology: a brief overview. *Science (New York, N.Y)*, 295:1662–4.

Lee, C. and Othmer, H. (2010). A multi-time-scale analysis of chemical reaction networks: I. deterministic systems. *Journal of mathematical biology*, 60(3):387–450.

Leisner, M., Kuhr, J., Rädler, J., Frey, E., and Maier, B. (2009). Kinetics of genetic switching into the state of bacterial competence. *Biophysical journal*, 96(3):1178–88.

Leisner, M., Stingl, K., Rädler, J., and Maier, B. (2007a). Basal expression rate of comk sets a 'switching-window' into the k-state of bacillus subtilis. *Molecular microbiology*, 63(6):1806–16.

Leisner, M., Stingl, K., Rädler, J., and Maier, B. (2007b). Basal expression rate of comk sets a 'switching-window' into the k-state of bacillus subtilis. *Molecular Microbiology*, 63(6):1806–1816.

Lippincott-Schwartz, G. and Patterson, G. (2003). Development and use of fluorescent protein markers in living cells. *Science (New York, N.Y.)*, 300:87–91.

Locker, J. (2000). *Transcription Factors*. BIOS Scientific Publishers Ltd.

López-Maury, L., Marguerat, S., and B´ahler, J. (2008). Tuning gene expression to changing environments: from rapid responses to evolutionary adaptation. *Nature reviews. Genetics*, 9(8):583–93.

Maamar, H. and Dubnau, D. (2005). Bistability in the bacillus subtilis k-state (competence) system requires a positive feedback loop. *Molecular Microbiology*, 56(3):615–624.

Maamar, H., Raj, A., and Dubnau, D. (2007). Noise in gene expression determines cell fate in bacillus subtilis. *Science (New York, N.Y)*, 317(5837).

Mehta, P., Mukhopadhyay, R., and Wingreen, N. (2008). Exponential sensitivity of noise-driven switching in genetic networks. *Physical biology*, 5:026005.

Pakka, V., Pr"ugel-Bennett, A., and Dasmahapatra, S. (2010). Correlated fluctuations carry signatures of gene regulatory network dynamics. *Journal of theoretical biology*, 266(3):343–57.

Phillips, T. (2008). Regulation of transcription and gene expression in eukaryotes. *Nature Education*, 1(1):199.

Press, W. H., Vetterling, W. T., and Plannery, B. P. (1992). *Numerical Recipes in C - The Art of Scientific Computing*. the Press Syndicate of the University of Cambridge, The Pitt Building, Trumpington street, Cambridge CB2 1RP, 2 edition.

Ralston, A. (2008). Simultaneous gene transcription and translation in bacteria. *Nature Education*, 1(1):4.

Ramsey, S., Orrell, D., and Bolouri, H. (2005). Dizzy:: Stochastic simulation of large-scale genetic regulatory networks. *Journal of Bioinformatics & Computational Biology*, 3(2):415–436.

Robinson, A. and van Oijen, A. (2013). Bacterial replication, transcription and translation: mechanistic insights from single-molecule biochemical studies. *Nature reviews. Microbiology*, 11(5):303–15.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.

Roussel, M. (1997). Forced-convergence iterative schemes for the approximation of invariant manifolds. *Journal of Mathematical Chemistry*, 21:385–393.

Roussel, M. and Fraser, S. (1990). Geometry of the steady-state approximation: Perturbation and accelerated convergence methods. *The Journal of Chemical Physics*, 93(2):1072.

Roussel, M. and Fraser, S. (2001). Invariant maniford methods for metabolic model reduction. *Chaos (Woodbury, N.Y.)*, 11(1):196–206.

Schrer, G. and Trenkler, D. (1995). Exact and randomization distributions of kolmogorov-smirnov tests two or three samples. *Computational Stat*, 20:185–202.

Shimomura, O., Johnson, F., and Saiga, Y. (1962). Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, aequorea. *Journal of Cellular & Comparative Physiology*, 59(3):223–239.

Sobolev, V. and Tropkina, E. (2012). Asymptotic expansions of slow invariant manifolds and reduction of chemical kinetics models. *Computational Mathematics and Mathematical Physics*, 52(1):75–89.

Sonenberg, N. and Hinnebusch, A. G. (2009). Regulation of translation initiation in eukaryotes: mechanisms and biological targets. *Cell*, 136(4):731–45.

Süel, G., Garcia-Ojalvo, J., Liberman, L., and Elowitz, M. (2006). An excitable gene regulatory circuit induces transient cellular differentiation. *Nature*, 440(7083):545–550.

Süel, G., Kulkarni, R., Dworkin, J., J., G.-O., and Elowitz, M. (2007). Tunability and noise dependence in differentiation dynamics. *Science (New York, N.Y.)*, 315(5819):1716–9.

Tao, Y., Zheng, X., and Sun, Y. (2007). Effect of feedback regulation on stochastic gene expression. *Journal of theoretical biology*, 247(4):827–36.

Tian, H. (2003). Asymptotic expansion for the solution of singularly perturbed delay differential equations. *Journal of Mathematical Analysis and Applications*, 281(2):678–696.

Turgay, K., Hahn, J., Burghoorn, J., and Dubnau, D. (1998). Competence in bacillus subtilis is controlled by regulated proteolysis of a transcription factor. *The EMBO journal*, 17(22):6730–8.

Veening, J., Smits, W., and Kuipers, O. (2008). Bistability, epigenetics, and bet-hedging in bacteria. *Annual review of microbiology*, 62:193–210.

White, R. J. (2001). *Gene Transcription : Mechanisms and Control*. Wiley-Blackwell, Hoboken, NJ, USA.

Zagaris, A., Kaper, H., and Kaper, T. (2004). Analysis of the computational singular perturbation reduction method for chemical kinetics. *Journal of Nonlinear Science*, 14(1):59–91.

Zeitlinger, J. and Stark, A. (2010). Developmental gene regulation in the era of genomics. *Developmental biology*, 339(2):230–9.