Chapter 9

Stochastic Hybrid Modeling of Biochemical Processes

Panagiotis Kouretas
University of Patras

Konstantinos Koutroumpas
ETH Zürich

John Lygeros
ETH Zürich

Zoi Lygerou
University of Patras

9.1 Introduction

One of the defining changes in molecular biology over the last decade has been the massive scaling up of its experimental techniques. The sequencing of the entire genome of organisms, the determination of the expression level of genes in a cell by means of DNA micro-arrays, and the identification of proteins and their interactions by high-throughput proteomic methods have produced enormous amounts of data on different aspects of the development and functioning of cells.

A consensus is now emerging among biologists that to exploit these data to full potential one needs to complement experimental results with formal models of biochemical networks. Mathematical models that describe gene and protein interactions in a precise and unambiguous manner can play an instrumental role in shaping the future of biology. For example, mathematical models allow computer-based simulation and analysis of biochemical networks. Such *in silico* experiments can be used for massive and rapid verification or falsification of biological hypotheses, replacing in certain cases costly and time-consuming in vitro or *in vivo* experiments. Moreover, *in silico*, *in vitro* and *in vivo* experiments can be used together in a feedback arrangement: mathematical model predictions can assist in the design of *in vitro* and *in vivo* experiments, the results of which can in turn be used to improve the fidelity of the mathematical models.

The possibility of combining new experimental methods, sophisticated mathematical techniques, and increasingly powerful computers, has given a new lease of life to an idea as appealing as it is difficult to realize: understanding how the global behavior of an organism emerges from the interactions between components at the molecular level. Although this idea of systems biology has multiple aspects [23], an ultimate challenge is the construction of a mathematical model of whole cells, that will be able to simulate *in silico* the behavior of an organism *in vivo*.

In the last few decades, a large number of approaches for modeling molecular interaction networks have been proposed. Motivated by the classification of [12], one can divide the models available in the literature in two classes:

- Models with purely continuous dynamics, for example, models that describe the evolution of concentrations of proteins, mRNAs, etc., in terms of ordinary or partial differential equations.
- Models with purely discrete dynamics, for example, graph models of the interdependencies in a regulatory network, Boolean networks and their extensions, Bayesian networks, or Markov chain models.

Common sense and experimental evidence suggest that neither of these classes alone is adequate for developing realistic models of molecular interaction networks. Timescale hierarchies cause biological processes to be more conveniently described as a mixture of continuous and discrete phenomena. For example, continuous changes in chemical concentrations or the environment of a cell often trigger discrete transitions (such as the onset of mitosis, or cell differentiation) that in turn influence the concentration dynamics. At the level of molecular interactions, the co-occurrence of discrete and continuous dynamics is exemplified by the switch-like activation or inhibition of gene expression by regulatory proteins.

The recognition that hybrid discrete-continuous dynamics can play an important role in biochemical systems has led a number of researchers to investigate how methods developed for hybrid systems in other areas (such as embedded computation and air traffic management) can be extended to biological systems [17, 1, 13, 5, 3, 15]. It is, however, fair to say that the realization of the potential of hybrid systems theory in the context of biochemical system modeling is still for the future. In addition, recently the observation that many biological processes involve considerable levels of uncertainty has been gaining momentum [27, 22]. For example, experimental observations suggest that stochastic uncertainty may play a crucial role in enhancing the robustness of biochemical processes [35], or may be behind the variability observed in the behavior of certain organisms [36, 37]. Stochasticity is even observed in fundamental processes such as the DNA replication itself [8, 26]. This has led researchers to attempt the development of stochastic hybrid models for certain biochemical processes [20, 19], that aim to couple the advantages of stochastic analysis with the generality of hybrid systems.

In this chapter we explore further stochastic, hybrid aspects in the modeling of biochemical networks. We first survey briefly a framework for modeling stochastic hybrid systems known as Piecewise Deterministic Markov Processes (PDMP; Section 9.2). We then proceed to use this framework to capture the essence of two biochemical processes: the production of subtilin by the bacterium *Bachillus subtilis* (*B. subtilis*; Section 9.3), and the process of DNA replication in eukaryotic cells (Section 9.4). The two models illustrate two different mechanisms through which stochastic features manifest themselves in biochemical processes: the uncertainty about switching genes "on" and "off" and uncertainty about the binding of protein complexes on the DNA. We also discuss how these models can be coded in simulation and present simulation results. The concluding section (Section 9.5) presents directions for further research.

9.2 Overview of PDMP

Piecewise Deterministic Markov Processes (PDMP), introduced by Mark Davis in [9, 10], are a class of continuous-time stochastic hybrid processes which covers a wide range of non-diffusion phenomena. They involve a hybrid state space, comprising continuous and discrete states. The defining feature of autonomous PDMP is that continuous motion is deterministic; between two consecutive transitions the continuous state evolves according to an ordinary differential equation (ODE). Transitions occur either when the state hits the state space boundary, or in the interior of the state space, according to a generalized Poisson process. Whenever a transition occurs, the hybrid state is reset instantaneously according to a probability distribution which depends on the hybrid state before the transition, and the process is repeated. Here we introduce formally PDMP following the notation of [6, 24]. Our treatment of PDMP is adequate for this chapter, but glosses over some of the technical subtleties introduced in [9, 10] to make the PDMP model as precise and general as possible.

9.2.1 Modeling Framework

Let Q be a countable set of discrete states, and let $d(\cdot) : Q \to \mathbb{N}$ and $X(\cdot) : Q \to \mathbb{R}^{d(\cdot)}$ be two maps assigning to each discrete state $q \in Q$ a continuous state dimension $d(q)$ and an open subset $X(q) \subseteq \mathbb{R}^{d(q)}$. We call the set

$$\mathscr{D}(Q,d,X) = \bigcup_{q\in Q} \{q\} \times X(q) = \{(q,x) \mid q \in Q,\ x \in X(q)\}$$

the hybrid state space of the PDMP and denote by $(q,x) \in \mathscr{D}(Q,d,X)$ the hybrid state. For simplicity, we use just $\mathscr{D}$ to denote the state space when the Q, d, and X

are clear from the context. We denote the complement of the hybrid state space by

$$\mathscr{D}^c = \bigcup_{q \in Q} \{q\} \times X(q)^c,$$

its closure by

$$\overline{\mathscr{D}} = \bigcup_{q \in Q} \{q\} \times \overline{X(q)},$$

and its boundary by

$$\partial\mathscr{D} = \bigcup_{q \in Q} \{q\} \times \partial X(q) = \overline{\mathscr{D}} \setminus \mathscr{D}.$$

As usual, $X(q)^c$ denotes the complement, $\overline{X(q)}$ the closure, and $\partial X(q)$ the boundary of the open set $X(q)$ in $\mathbb{R}^{d(q)}$, and $\setminus$ denotes set difference. Let $\mathscr{B}(\mathscr{D})$ denote the smallest σ-algebra on $\cup_{q\in Q}\{q\} \times \mathbb{R}^{d(q)}$ containing all sets of the form $\{q\} \times A_q$ with A_q a Borel subset of $X(q)$.

We consider a parameterized family of vector fields $f(q,\cdot) : \mathbb{R}^{d(q)} \to \mathbb{R}^{d(q)}$, $q \in Q$, assigning to each hybrid state (q,x) a direction $f(q,x) \in \mathbb{R}^{d(q)}$. As usual, we define the flow of f as the function $\Phi(q,\cdot,\cdot) : \mathbb{R}^{d(q)} \times \mathbb{R} \to \mathbb{R}^{d(q)}$ such that $\Phi(q,x,0) = x$ and for all $t \in \mathbb{R}$,

$$\frac{d}{dt}\Phi(q,x,t) = f(q,\Phi(q,x,t)). \tag{9.1}$$

Notice that we implicitly assume that the discrete state q remains constant along continuous evolution.

DEFINITION 9.1 *A PDMP is a collection $H = ((Q,d,X), f, Init, \lambda, R)$, where*

- *Q is a countable set of discrete states;*
- *$d(\cdot) : Q \to \mathbb{N}$ maps each $q \in Q$ to a continuous state space dimension;*
- *$X(\cdot) : Q \to \mathbb{R}^{d(\cdot)}$ maps each $q \in Q$ to an open subset $X(q)$ of $\mathbb{R}^{d(q)}$;*
- *$f(q,\cdot) : \mathbb{R}^{d(q)} \to \mathbb{R}^{d(q)}$ is a family of vector fields parameterized by $q \in Q$;*
- *$\mathrm{Init}(\cdot) : \mathscr{B}(\mathscr{D}) \to [0,1]$ is an initial probability measure on $(\mathscr{D}, \mathscr{B}(\mathscr{D}))$;*
- *$\lambda(\cdot,\cdot) : \mathscr{D} \to \mathbb{R}^+$ is a transition rate function;*
- *$R(\cdot,\cdot,\cdot) : \mathscr{B}(\mathscr{D}) \times \overline{\mathscr{D}} \to [0,1]$ assigns to each $(q,x) \in \overline{\mathscr{D}}$ a measure $R(\cdot,q,x)$ on $(\mathscr{D}, \mathscr{B}(\mathscr{D}))$.*

To define the PDMP executions we introduce the notions of the exit time, $t^*(\cdot,\cdot) : \mathscr{D} \to \mathbb{R}^+ \cup \{\infty\}$, defined as

$$t^*(q,x) = \inf\{t > 0 \mid \Phi(q,x,t) \notin \mathscr{D}\} \tag{9.2}$$

and of the survival function $F(\cdot,\cdot,\cdot) : \mathscr{D} \times \mathbb{R}^+ \to [0,1]$,

$$F(q,x,t) = \begin{cases} e^{-\int_0^t \lambda(q,\Phi(q,x,\tau))d\tau} & \text{if } t < t^*(q,x) \\ 0 & \text{if } t \geq t^*(q,x). \end{cases} \tag{9.3}$$

With this notation, the executions of the PDMP can be thought of as being generated by the following algorithm.

ALGORITHM 9.1 **(Generation of PDMP executions)**

```
set $T = 0$
extract $\mathscr{D}$-valued random variable $(\hat{q},\hat{x})$ according to Init$(\cdot)$
while $T < \infty$
    extract $\mathbb{R}^+$-valued random variable $\hat{T}$ such that $P[\hat{T} > t] = F(\hat{q},\hat{x},t)$
    set $q(t) = \hat{q}$ and $x(t) = \Phi(\hat{q},\hat{x},t-T)$ for all $t \in [T, T+\hat{T})$
    if $\hat{T} < \infty$
        extract $\mathscr{D}$-valued random variable $(q',x')$
            according to $R((\cdot,\cdot,\hat{q},\Phi(\hat{q},\hat{x},\hat{T}))$
        set $(\hat{q},\hat{x}) = (q',x')$
    end if
    set $T = T + \hat{T}$
end while
```

All random extractions in Algorithm 9.1 are assumed to be independent. To ensure that the algorithm produces a well-defined stochastic process a number of assumptions are introduced in [10, 9].

ASSUMPTION 9.1 The PDMP satisfies the following assumptions:

(i) *Init*$(\mathscr{D}^c) = 0$ and $R(\mathscr{D}^c,q,x) = 0$ for all $(q,x) \in \overline{\mathscr{D}}$.

(ii) For all $q \in Q$, the set $X(q) \subseteq \mathbb{R}^{d(q)}$ is open and $f(q,\cdot)$ is globally Lipschitz continuous.

(iii) $\lambda(\cdot,\cdot)$ is measurable. For all $(q,x) \in \mathscr{D}$ there exists $\varepsilon > 0$ such that the function $t \to \lambda(q,\Phi(q,x,t))$ is integrable for $t \in [0,\varepsilon)$. For all $A \in \mathscr{B}(\mathscr{D})$, $R(A,(\cdot,\cdot))$ is measurable.

(iv) The expected number of jumps in $[0,t]$ is finite for all $t < \infty$.

Most of the assumptions are technical and are needed to ensure that the transition kernels, the solutions of the differential equations, etc., are well defined. The last part of the assumption deserves some closer scrutiny. This is the stochastic variant of the non-Zeno assumption commonly imposed on hybrid systems. It states that "on the average" only a finite number of discrete transitions can take place in any finite time interval. While this assumption is generally true for real systems, it is easy to generate models that violate it due to modeling over-abstraction (see, for example, [21]). Even if a model is not Zeno, establishing this this may be difficult [25].

Under Assumption 9.1 it can be shown [10, 9] that Algorithm 9.1 defines a strong Markov process, which is continuous from the right with left limits. Based on these fundamental properties, [10, 9] proceed to completely characterize PDMP processes through their generator, and then use the generator to show how one can compute expectations, establish stability conditions and solve optimal control problems for this class of stochastic hybrid systems.

9.2.2 Simulation

The properties of PDMP are in general rather difficult to study analytically. Explicit solutions for things like expectations are impossible to derive, except in very special cases (see, for example, [11]). One therefore often resorts to numerical methods. For computing expectations, approximating distributions, etc., one of the most popular is Monte Carlo simulation.

The simulation of PDMP models presents several challenges, due to the interaction of discrete, continuous, and stochastic terms. Because the continuous dynamics are deterministic, standard algorithms used for the simulation of continuous, deterministic systems are adequate for simulating the evolution between two discrete transitions. The difficulties arise when the continuous evolution has to be interrupted so that a discrete transition can be executed.

For forced transitions (when the state attempts to leave $\mathscr{D}$) one needs to detect when the state, x, leaves an open set, $X(q)$, along continuous evolution. This is known as the event detection problem in the hybrid systems literature. Several algorithms have been developed to deal with this problem (see for example [4]) and have recently been included in standard simulation packages such as Dymola, Matlab, or the Simulink package SimEvents. Roughly speaking, the idea is to code the set $X(q)$ using a function, $g(q,x)$, of the state that changes sign at the boundary of $X(q)$. The simulation algorithm keeps track of the function $g(q,x(k))$ at each step, k, of the continuous simulation and proceeds normally as long as $g(q,x(k))$ does not change sign between one step and the next; recall that in this case q also does not change. If g changes sign (say between step k and step $k+1$) the simulation halts, a zero crossing algorithm is used to pinpoint the time at which the sign change took place, the state at this time is computed, the event is "serviced," and the simulation resumes from the new state. Zero crossing (finding the precise state just before the event) usually involves fitting a polynomial to a few values of g before and after the event (say a spline through $g(q,x(k-1))$, $g(q,x(k))$ and $g(q,x(k+1))$) and finding its roots. Servicing the event (finding the state just after the event) requires a random extraction from the transition kernel R. While it is known that for most hybrid systems initial conditions exist for which accurate event detection is problematic, it is also known that for a wide class of hybrid systems the simulation strategy outlined above works for almost all initial states [33].

For spontaneous transitions, the situation is at first sight more difficult: one needs to extract a random transition time, $\hat{T}$, such that

$$P[\hat{T} > t] = e^{-\int_0^t \lambda(\hat{q},\Phi(\hat{q},\hat{x},\tau))d\tau}.$$

It turns out, however, that this can easily be done by appending two additional continuous states, say $y \in \mathbb{R}$ and $z \in \mathbb{R}$, to the state vector. We therefore make a new PDMP, $H' = ((Q, d', X'), f', Init', \lambda', R')$ with continuous state dimension

$$d'(q) = d(q) + 2, \text{for all } q \in Q.$$

We set

$$X'(q) = X(q) \times \{(y,z) \in \mathbb{R}^2 \mid z > 0,\, y < -\ln(z)\}.$$

The continuous dynamics of these additional states are given by $\dot{y} = \lambda(q,x)$ and $\dot{z} = 0$, in other words

$$f'(q,x) = \begin{bmatrix} f(q,x) \\ \lambda(q,x) \\ 0 \end{bmatrix}.$$

We set $\lambda'(q,x,y,z) = 0$ for all $q \in Q$, $(x,y,z) \in \mathbb{R}^{d'(q)}$.

Initially, and after each discrete transition y is set to 0, whereas z is extracted uniformly in the interval $[0,1]$. For simplicity consider the first interval of continuous evolution; the same argument holds between any two transitions. Until the first discrete transition we have

$$y(t) = \int_0^t \lambda(q(0), \Phi(q(0), x(0), \tau))d\tau, \text{ and } z(t) = z(0).$$

Notice that, since λ is non-negative, the state $y(t)$ is a non-decreasing function of t.

Since λ' is identically equal to zero spontaneous transitions are not possible for the modified PDMP. Therefore the first transition will take place because either $x(t)$ leaves $X(q)$, or because $y(t) \geq -\ln(z(t))$. Assume the latter is the case, and let

$$\hat{T} = \inf\{\tau \geq 0 \mid y(\tau) \geq -\ln(z(\tau))\}.$$

Then

$$\begin{aligned} P[\hat{T} > t] &= P[y(t) < -\ln(z(t))] && (y(t) \text{ non-decreasing}) \\ &= P[\textstyle\int_0^t \lambda(q(0), \Phi(q(0), x(0), \tau))d\tau < -\ln(z(t))] \\ &= P[z(0) < e^{-\int_0^t \lambda(q(0), \Phi(q(0), x(0), \tau))d\tau}] \\ &= e^{-\int_0^t \lambda(q(0), \Phi(q(0), x(0), \tau))d\tau} && (z(0) \text{ uniform}). \end{aligned}$$

After the discrete transition the new state (q,x) is extracted according to R, y is reset to zero and z is extracted uniformly in $[0,1]$.

Therefore, for simulation purposes spontaneous transitions can be treated in very much the same way as forced transitions. In fact, the above construction is standard in the simulation of discrete event systems [7] and shows that every PDMP, H, is equivalent to another PDMP, H', that involves only forced transitions. Spontaneous transitions, however, still provide a very natural way of modeling physical phenomena and will be used extensively below.

9.3 Subtilin Production by *B. subtilis*

9.3.1 Qualitative Description

Subtilin is an antibiotic released by *B. subtilis* as a way of confronting difficult environmental conditions. The factors that govern subtilin production can be divided into internal (the physiological states of the cell) and external (local population density, nutrient levels, aeration, environmental signals, etc.). Roughly speaking, a high concentration of nutrients in the environment results in an increase in *B. subtilis* population without a remarkable change in subtilin concentration. Subtilin production starts when the amount of nutrient falls under a threshold because of excessive population growth [29]. *B. subtilis* then produces subtilin and uses it as a weapon to increase its food supply, by eliminating competing species; in addition to reducing the demand for nutrients, the decomposition of the organisms killed by subtilin releases additional nutrients in the environment.

According to the simplified model for the subtilin production process developed in [20], subtilin derives from the peptide SpaS. Responsible for the production of SpaS is the activated protein SpaRK, which in turn is produced by the SigH protein. Finally, the composition of SigH is turned on whenever the nutrient concentration falls below a certain threshold.

9.3.2 An Initial Model

An initial stochastic hybrid model for this process was proposed in [20]. The model comprises 5 continuous states: the population of *B. subtilis*, x_1, the concentration of nutrients in the environment, x_2, and the concentrations of the SigH, SpaRK and SpaS molecules (x_3, x_4, and x_5 respectively). The model also comprises $2^3 = 8$ discrete states, generated by three binary switches, which we denote by S_3, S_4 and S_5. Switch S_3 is deterministic: it goes ON when the concentration of nutrients, x_2, falls below a certain threshold (denoted by η), and OFF when it rises over this threshold. The other two switches are stochastic. In [20] this stochastic behavior is approximated by a discrete time Markov chain, with constant sampling interval Δ. Given that the switch S_4 is OFF at time $k\Delta$, the probability that it will be ON at time $(k+1)\Delta$ depends on the concentration of SigH at the time $k\Delta$ and is given by

$$a_0(x_3) = \frac{cx_3}{1+cx_3}, \tag{9.4}$$

The nonlinear form of this equation is common for chemical reactions, such as the activation of genes, that involve “binding” of proteins to the DNA. Roughly speaking, the higher x_3 is the more SigH molecules are around and the higher the probability that one of them will bind with the DNA activating the gene that produces SpaRK. The constant c is a model parameter that depends on the activation energy of the reaction (reflecting the natural “propensity” of the particular molecules to bind) and the temperature. It will be shown below that $x_3 \geq 0$ (as expected for a concentration) therefore $a_0(x_3)$ can indeed be thought of as a probability. Notice that the probability of switching ON increases to 1 as x_3 gets higher. Conversely, given that the switch S_4 is ON at time $k\Delta$, the probability that it will be OFF at time $(k+1)\Delta$ is

$$a_1(x_3) = 1 - a_0(x_3) = \frac{1}{1+cx_3}. \tag{9.5}$$

Notice that this probability increases to 1 as x_3 gets smaller. The dynamics of switch S_5 are similar, with the concentration of SpaRK, x_4, replacing x_3 and a different value, c', for the constant.

The dynamics for the *B. subtilis* population x_1 are given by the logistic equation

$$\dot{x}_1 = rx_1(1 - \frac{x_1}{D_\infty(x_2)}). \tag{9.6}$$

Under this equation, x_1 will tend to converge to

$$D_\infty(x_2) = \min\{\frac{x_2}{X_0}, D_{\max}\}, \tag{9.7}$$

the steady state population for a given nutrient amount. X_0 and $D_{\max}$ are constants of the model; the latter represents constraints on the population because of space limitations and competition within the population.

The dynamics for x_2 are governed by

$$\dot{x}_2 = -k_1x_1 + k_2x_5 \tag{9.8}$$

where k_1 denotes the rate of nutrient consumption per unit of population and k_2 the rate of nutrient production due to the action of subtilin. More precisely, the second term is proportional to the average concentration of SpaS, but for simplicity [20] assume that the average concentration is proportional to the concentration of SpaS for a single cell.

The dynamics for the remaining three states depend on the discrete state, i.e., the state of the three switches. In all three cases,

$$\dot{x}_i = \begin{cases} -l_ix_i & \text{if } S_i \text{ is OFF} \\ k_i - l_ix_i & \text{if } S_i \text{ is ON.} \end{cases} \tag{9.9}$$

It is easy to see that the concentration x_i decreases exponentially toward zero whenever the switch S_i is OFF and tends exponentially toward k_i/l_i whenever S_i is ON. Note that the model is closely related to the piecewise affine models studied by [13, 5]. The key differences are the nonlinear dynamics of x_1 and the stochastic terms used to describe the switch behavior.

9.3.3 A Formal PDMP Model

We now try to develop a PDMP, $H = ((Q,d,X), f, Init, \lambda, R)$, to capture the mechanism behind subtilin production outlined above. To do this we need to define all the quantities listed in Definition 9.1.

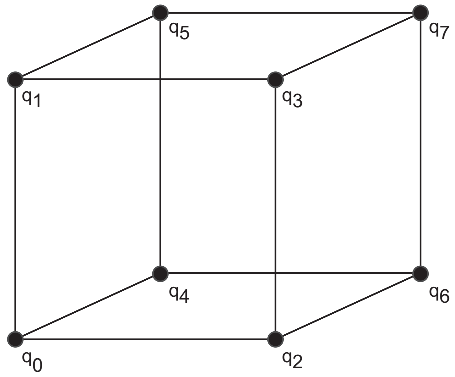

FIGURE 9.1: Visualization of discrete state space.

The presence of the three switches (S_3, S_4, and S_5) dictate that the PDMP model should have 8 discrete states (see Figure 9.1). We denote these discrete states by

$$Q = \{q_0, \ldots, q_7\}, \tag{9.10}$$

so that the index (in binary) of each discrete state reflects the state of the three switches. For example, state q_0 corresponds to binary 000, i.e., all three switches being OFF. Likewise, state q_5 corresponds to binary 101, i.e., switches S_3 and S_5 being ON and switch S_4 being OFF. In the following discussion, the state names $q_0, \ldots, q_7$ and the binary equivalents of their indices will be used interchangeably. A wildcard, $*$, will be used when in a statement the position of some switch is immaterial; e.g., $1**$ denotes that something holds when S_3 is ON, whatever the values of S_4 and S_5 may be.

The discussion in the previous section suggests that there are 5 continuous states and all of them are active in all discrete states. Therefore, the dimension of the continuous state space is constant

$$d(q) = 5, \text{ for all } q \in Q. \tag{9.11}$$

The definition of the survival function (9.3) suggests that the open sets $X(q) \subseteq \mathbb{R}^5$ are used to force discrete transitions to take place at certain values of state. In the subtlin production model outlined above the only forced transitions are those induced by the deterministic switch S_3: S_3 has to go ON whenever x_2 falls under the threshold η and has to go OFF whenever it rises over this threshold. These transitions can be forced by defining

$$X(0**) = \mathbb{R} \times (\eta, \infty) \times \mathbb{R}^3 \text{ and } X(1**) = \mathbb{R} \times (-\infty, \eta) \times \mathbb{R}^3. \tag{9.12}$$

The three elements defined in Equations (9.10)–(9.12) completely determine the hybrid state space, $\mathscr{D}(Q,d,X)$, of the PDMP.

The family of vector fields, $f(q,\cdot)$, is easy to infer from the above discussion. From Equations (9.6)–(9.9) we have that

$$f(q_0,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ -l_3x_3 \\ -l_4x_4 \\ -l_5x_5 \end{bmatrix} \quad f(q_1,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ -l_3x_3 \\ -l_4x_4 \\ k_5-l_5x_5 \end{bmatrix}$$

$$f(q_2,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ -l_3x_3 \\ k_4-l_4x_4 \\ -l_5x_5 \end{bmatrix} \quad f(q_3,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ -l_3x_3 \\ k_4-l_4x_4 \\ k_5-l_5x_5 \end{bmatrix}$$

$$f(q_4,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ k_3-l_3x_3 \\ -l_4x_4 \\ -l_5x_5 \end{bmatrix} \quad f(q_5,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ k_3-l_3x_3 \\ -l_4x_4 \\ k_5-l_5x_5 \end{bmatrix}$$

$$f(q_6,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ k_3-l_3x_3 \\ k_4-l_4x_4 \\ -l_5x_5 \end{bmatrix} \quad f(q_7,x)=\begin{bmatrix} rx_1(1-\frac{x_1}{\min\{x_2/X_0,D_{\max}\}}) \\ -k_1x_1+k_2x_3 \\ k_3-l_3x_3 \\ k_4-l_4x_4 \\ k_5-l_5x_i \end{bmatrix}.$$

Notice that most of the equations are affine (and hence globally Lipschitz) in x. The only difficulty may arise from the population equation which is nonlinear. However, the bounds on x_1 and x_2 established in Proposition 9.1 below ensure that Assumption 9.1 is met.

The probability distribution, *Init*, for the initial state of the model should respect the constraints imposed by Assumption 9.1. We therefore require that the distribution satisfies

$$Init(\{0**\}\times\{x\in\mathbb{R}^5 \mid x_2\le\eta\})=0,\ Init(\{1**\}\times\{x\in\mathbb{R}^5 \mid x_2\ge\eta\})=0. \quad (9.13)$$

The initial state should also reflect any other constraints imposed by biological intuition. For example, since x_1 reflects the *B. subtilis* population, it is reasonable to assume that $x_1 \geq 0$ (at least initially). Moreover, the form of the logistic equation (9.6) suggests that another reasonable constraint is that initially $x_1 \leq D_\infty(x_2)$. Finally, since continuous states $x_2,\ldots,x_5$ reflect concentrations, it is reasonable to assume that they also start with non-negative values. These constraints can be imposed if we require that for all $q \in Q$

$$Init(\{q\}\times\{x\in\mathbb{R}^5 \mid x_1\in(0,D_\infty(x_2)) \text{ and } \min\{x_2,x_3,x_4,x_5\}>0\})=1. \quad (9.14)$$

Any probability distribution that respects constraints (9.13) and (9.14) is an acceptable initial state probability distribution for our model.

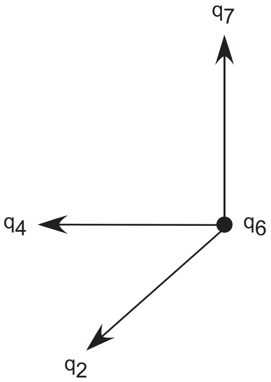

FIGURE 9.2: Transitions out of state q_6.

The main problem we confront when trying to express the subtilin production model as a PDMP is the definition of the rate function λ. Intuitively, this function indicates the "tendency" of the system to switch its discrete state. The rate function λ will govern the spontaneous transitions of the switches S_4 and S_5; recall that S_3 is deterministic and is governed by forced transitions. To present the design of an appropriate function λ we focus on discrete state q_6 (the design of λ for the other discrete states is similar). Figure 9.2 summarizes the discrete transitions out of state q_6. Notice that simultaneous switching of more than one of the switches S_3, S_4, S_5 is not allowed. This makes the PDMP model of the system more streamlined. It is also a reasonable assumption to make, since simultaneous switching of two or more switches is a null event in the underlying probability space.

Recall that q_6 corresponds to binary 110, i.e., switches S_3 and S_4 being ON and S_5 being OFF. Of the three transitions out of q_6, the one to q_2 ($S_3 \rightarrow$ OFF) is forced and does not feature in the construction of the rate function λ. For the remaining two transitions, we define two separate rate functions, $\lambda_{S_4 \rightarrow \text{ OFF}}(x)$ and $\lambda_{S_5 \rightarrow \text{ ON}}(x)$. These functions need to be linked somehow to the transition probabilities of the discrete time Markov chain with sampling period Δ used to model the probabilistic switching in [20]. This can be done via the survival function of Equation (9.3). The survival function states that the probability that the switch S_4 remains ON throughout the interval $[k\Delta, (k+1)\Delta]$ is equal to

$$\exp\left(-\int_{k\Delta}^{(k+1)\Delta} \lambda_{S_4 \rightarrow \text{ OFF}}(x(\tau))d\tau\right).$$

According to Equation (9.4) this probability should be equal to $1 - a_0(x_3(k\Delta))$. Assuming that Δ is small enough, we have that

$$1 - a_0(x_3(k\Delta)) \approx \exp\left(-\Delta\lambda_{S_4 \rightarrow \text{ OFF}}(x(k\Delta))\right).$$

Selecting

$$\lambda_{S_4 \to \text{ OFF}}(x) = \frac{\ln(1 + cx_3)}{\Delta} \tag{9.15}$$

achieves the desired effect. Likewise, we define

$$\lambda_{S_5 \to \text{ ON}}(x) = \frac{\ln(1 + c'x_4) - \ln(c'x_4)}{\Delta} \tag{9.16}$$

and set the transition rate for discrete state q_6 to

$$\lambda(q_6, x) = \lambda_{S_4 \to \text{ OFF}}(x) + \lambda_{S_5 \to \text{ ON}}(x). \tag{9.17}$$

Notice that the functions $\lambda_{S_4 \to \text{ OFF}}(x)$ and $\lambda_{S_5 \to \text{ ON}}(x)$ take non-negative values and are therefore good candidates for rate functions. $\lambda_{S_5 \to \text{ ON}}(x)$ is discontinuous at $x_3 = 0$, but the form of the vector field for x_3 ensures that there exists $\varepsilon > 0$ small enough such that if $x_3(0) > 0$, $\lambda_{S_5 \to \text{ ON}}(x(t))$ is integrable along the solutions of the differential equation over $t \in [0, \varepsilon)$. In a similar way, one can define rate functions $\lambda_{S_5 \to \text{ OFF}}(x)$ (replacing x_3 by x_4 and c by c' in Equation (9.15)) and $\lambda_{S_4 \to \text{ ON}}(x)$ (replacing x_4 by x_3 and c' by c in Equation (9.16)) and use them to define the transition rates for the remaining discrete states (in a way analogous to Equation (9.17)).

The last thing we need to define to complete the PDMP model is the probability distribution for the state after a discrete transition. The only difficulty here is removing any ambiguities that may be caused by simultaneous switches of two or more of S_3, S_4, and S_5. We do this by introducing a priority scheme: Whenever the forced transition has to take place it does, else either of the spontaneous transitions can take place. For state q_6 this leads to

$$R(q_6, x) = \delta_{(q_2, x)}(q, x) \text{ if } (q_6, x) \in \mathscr{D}^c \tag{9.18}$$

else

$$R(q_6, x) = \frac{\lambda_{S_4 \to \text{ OFF}}(x)}{\lambda(q_6, x)} \delta_{(q_4, x)}(q, x) + \frac{\lambda_{S_5 \to \text{ ON}}(x)}{\lambda(q_6, x)} \delta_{(q_7, x)}(q, x). \tag{9.19}$$

Here $\delta_{(\hat{q}, \hat{x})}(q, x)$ denotes the Dirac measure concentrated at $(\hat{q}, \hat{x})$. If desired, the two components of the measure R ((9.18) corresponding to the forced transition and (9.19) corresponding to the spontaneous transitions) can be written together using the indicator function, $I_{\mathscr{D}}(q, x)$, of the set $\mathscr{D}$.

$$\begin{aligned} R(q_6, x) = {} & (1 - I_{\mathscr{D}}(q_6, x))\, \delta_{(q_2, x)}(q, x) + \\ & I_{\mathscr{D}}(q_6, x) \left(\tfrac{\lambda_{S_4 \to \text{ OFF}}(x)}{\lambda(q_6, x)} \delta_{(q_4, x)}(q, x) + \tfrac{\lambda_{S_5 \to \text{ ON}}(x)}{\lambda(q_6, x)} \delta_{(q_7, x)}(q, x) \right). \end{aligned}$$

The measure R for the other discrete states can be defined in an analogous manner. It is easy to see that this probability measure satisfies Assumption 9.1.

The above discussion shows that the PDMP model also satisfies most of the conditions of Assumption 9.1. The only problem may be the non-Zeno condition. While this condition is likely to hold because of the structure of the vector fields and the transition rates, showing theoretically that it does is quite challenging.

9.3.4 Analysis and Simulation

To ensure that the model makes biological sense and to simplify somewhat the analysis, we impose the following restrictions on the values of the parameters.

ASSUMPTION 9.2 All model constants c, η, r, X_0, $D_{\max}$, k_i for $i=1,\ldots,5$ and l_i for $i=3,4,5$ are positive. Moreover, $k_1 < rX_0$.

Under these assumptions the following fact is easy to establish:

PROPOSITION 9.1 *Almost surely:*

(i) For all $t \geq 0$, $(q(t), x(t)) \in \overline{\mathscr{D}}$ and for almost all $t \geq 0$, $(q(t), x(t)) \in \mathscr{D}$.

(ii) For all $t \geq 0$, $x_1(t) \in [0, D_\infty(x_2(t))]$, and $\min\{x_2(t), x_3(t), x_4(t), x_5(t)\} > 0$.

PROOF (Outline) The first part is a general property of PDMP, and follows directly from (9.13). For the second part, the proof can be done by induction. We note first that by (9.14), the conditions hold almost surely for the initial state. The discrete transitions leave the continuous state unaffected, therefore we only have to show that the conditions remain valid along continuous evolution.

Let $x(0)$ denote the state at the beginning of an interval of continuous evolution and assume that condition 2 holds for this state. The form of the vector field is such $\dot{x}_3 \geq -l_3 x_3$. Therefore, $x_3(t) \geq e^{-l_3 t} x_3(0) > 0$ throughout the interval of continuous evolution. Similar arguments show that $x_4(t) > 0$ and $x_5(t) > 0$. Moreover, since $\dot{x}_1 = 0$ if $x_1 = 0$ or $x_1 = D_\infty(x_2)$, $x_1(t)$ remains in the interval $[0, D_\infty(x_2)]$ if it is initially in this interval.

It remains to show that $x_2(t) > 0$. Consider the function $V(x) = \frac{x_1}{x_2}$. Differentiating and assuming $x_2 \leq D_{\max}/X_0$ we get

$$\begin{aligned}
\dot{V}(x) &= \tfrac{rx_1}{x_2}\left(1 - \tfrac{X_0 x_1}{x_2}\right) - \tfrac{x_1}{x_2^2}(-k_1 x_1 + k_2 x_3)\\
&= r\tfrac{x_1}{x_2} - (rX_0 - k_1)\left(\tfrac{x_1}{x_2}\right)^2 - k_2 \tfrac{x_1 x_3}{x_2^2}\\
&\leq r\tfrac{x_1}{x_2} - (rX_0 - k_1)\left(\tfrac{x_1}{x_2}\right)^2
\end{aligned}$$

If we let $\alpha = x_1/x_2$ the last inequality reads $\dot{\alpha} \leq r\alpha - (rX_0 - k_1)\alpha^2$. If $k_1 < rX_0$, then for α large enough (equivalently, x_2 small enough since x_1 is bounded) the quadratic term dominates and keeps α bounded. Therefore, $x_2(t) > 0$ along continuous evolution. ■

Even though some additional facts about this model can be established analytically, the most productive way to analyze the model (especially its stochastic behavior) is by simulation. The model can easily be coded in simulation, using the methods

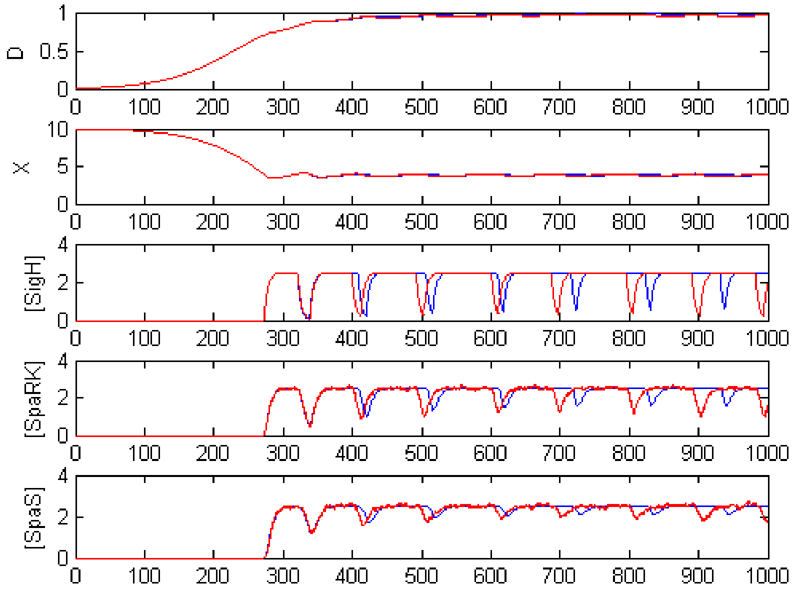

FIGURE 9.3: Sample solution for PDMP model of subtilin production.

outlined in Section 9.2.2. In this case the only forced transitions are those governing the switch S_3. An obvious choice for a function to code these forced transitions as zero crossings is $g(q,x) = x_2 - \eta$; the same function can be used for switching S_3 ON (crossing zero from above) and OFF (crossing zero from below). Servicing the event simply involves switching the state of S_3.

The model was coded in Matlab using ode45 with events enabled. Typical trajectories of the system is shown in Figure 9.3.

9.4 DNA Replication in the Cell Cycle

9.4.1 Qualitative Description

DNA replication, the process of duplication of the cells genetic material, is central to the life of every living cell, and is always carried out prior to cell division to ensure that the cells genetic information is maintained. Replication takes place during a specific stage in the life cycle of a cell (the cell cycle). The cell cycle (as shown in

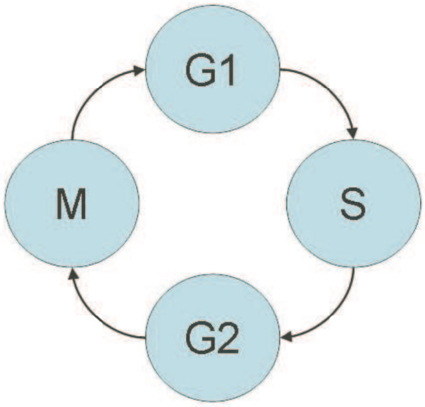

FIGURE 9.4: The phases of cell cycle.

Figure 9.4) can be subdivided in four phases: G1, a growth phase in which the cell increases its mass; S (synthesis), when DNA replication takes place; G2, a second growth phase; and finally M phase (mitosis), during which the cell divides and gives rise to two daughter cells.

Cell cycle events are regulated by the periodic fluctuations in the activity of protein complexes called Cyclin Dependent Kinases (CDK). CDK are the master regulators of the cell cycle [30]. In Figure 9.5, the so-called quantitative model of cell cycle regulation is illustrated. There are two identified thresholds in CDK activity, threshold 1 associated with entry into S phase and threshold 2 associated with entry into mitosis. Complex models have already been developed for the biochemical network regulating the fluctuation of CDK activity during the cell cycle [34].

Because daughter cells must have the same genetic information as their progenitor, during S-phase, every base of the genome must be replicated once and only once. Genomes of eukaryotic cells are large in size and the speed of replication is limited. To accelerate the process, DNA replication initiates from multiple points along the chromosomes, called origins of replication. Following initiation from a given origin, replication continues bi-directionally along the genome, giving rise to two replication forks moving in opposite directions.

To be able to ensure that each region of the genome is replicated once and only once, a cell must be able to distinguish a replicated from an unreplicated region. Before replication, and while CDK activity is low, origins are present in the pre-replicative state and can initiate DNA replication when CDK activity passes threshold 1. When an origin fires (or when it is passively replicated by a passing replication fork from a nearby origin) it automatically switches to the post-replicating state and can no longer support initiation of replication. CDK activities over threshold 1 inhibit conversion of the post-replicating state to the pre-replicative state. To re-acquire the

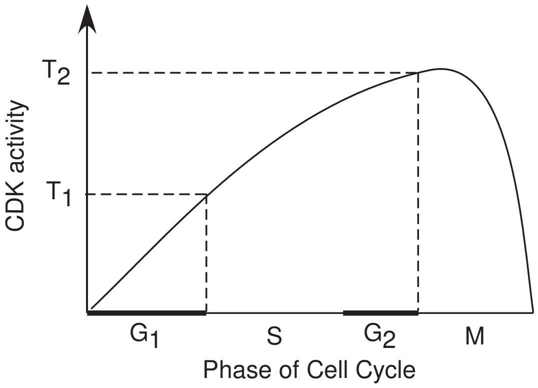

FIGURE 9.5: Quantitative model of cell cycle regulation.

pre-replicative state origins must wait for the end of the M phase, when CDK activity resets to zero. With this simple mechanism, re-replication is prevented.

Initial models, influenced by the replication of bacterial genomes, postulated that defined regions in the genome would act as origins of replication in every cell cycle [16]. Indeed, initial work from the budding yeast (*Saccharomyces cerevisiae*) identified specific sequences which acted as origins of replication with high efficiency [31]. This simple deterministic model of origin selection however is reappraised following more recent findings which show that, especially in higher eukaryotes, a large number of potential origins exist, and active origins are stochastically selected during each S phase [8, 26]. For example, recent work on the fission yeast *Schizosaccharomyces pombe* [26] clearly showed that origins fire stochastically during the S phase. The fission yeast genome has many hundreds of potential origins, but only a few of them fire in any given cell cycle. Multicellular eukaryotes are also believed to exhibit similar behavior.

9.4.2 Stochastic Hybrid Features

There are two main sources of uncertainty in the DNA replication process. The first has to do with which origins of replication fire in a particular cell cycle and the second with the times at which they fire.

Not all the origins participate in every S phase [8, 26]. Origins are classified as *strong* and *weak*, according to the frequency with which they fire. Given a population of cells undergoing an S-phase, strong origins are observed to fire in many cells, whereas weak ones fire in only a few. This firing probability is typically encoded as a number between 0 and 1 for each origin, reflecting the percentage of cells in which the particular origin is observed to fire.

Even if an origin does fire, the time during the S phase when it will do so is still uncertain. Some origins have been observed to fire earlier in the S phase, while

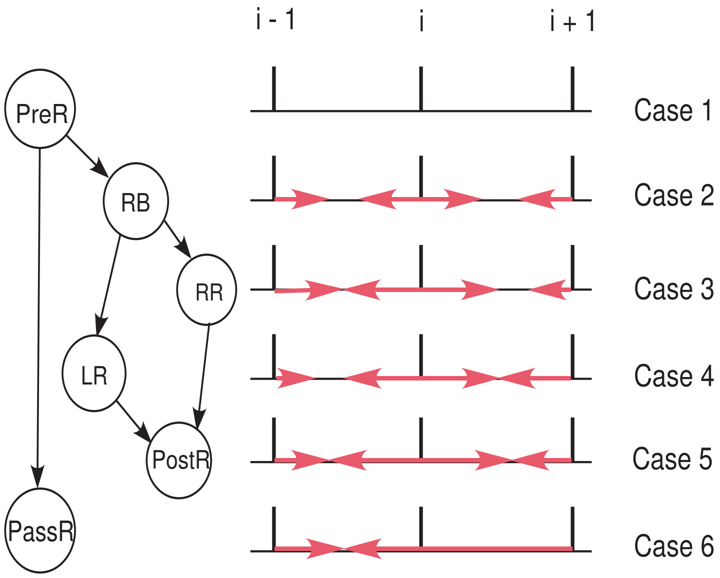

FIGURE 9.6: Possible states of an origin.

others tend to fire later [14]. This timing aspect is usually encoded by a number in minutes, reflecting the average firing time of the origin in a population of cells.

The two uncertainty parameters, efficiency and firing time, are clearly correlated. Late firing origins will also tend to have smaller efficiencies. This is because origins that tend to fire later during S phase give the chance to nearby early firing origins to replicate the part of the genome around them [18]. It is an on going debate among the cell cycle community as to whether these two manifestations of uncertainty are in fact one and the same, or whether there are separate biological mechanisms that determine if an origin is weak versus strong and early firing versus late firing. The hope is that mathematical models, like the one presented below, will assist biologists in answering such questions.

During the S phase an origin of replication may find itself in a number of "states". These are summarized in Figure 9.6, where we concentrate on an origin i and its neighbors, denoted by $i-1$ (left) and $i+1$ (right). We distinguish a number of cases.

Case 1: Pre-replicative. Every origin is in this mode until the time it becomes active (firing time). In this case it does not replicate in any direction.

Case 2: Replicating on both sides. When the origin firing time is reached, the origin gets activated and begins to replicate the DNA to its left and right. The points of replication ("forks") move away from the origin with a certain speed ("fork velocity").

Case 3: Right replicating. When the section of DNA that i has replicated to its left reaches the section of DNA that $i-1$ has replicated to its right, then the whole section between the two origins has been replicated. Origin i does not need to do any more replication to its left and so it continues only to the right.

Case 4: Left replicating. This is symmetric to Case 3. Origin i stops replication to the right and continues to the left.

Case 5: Post replicating. The replication has finished in both directions and the origin has completed its job.

Case 6: Passively replicated. The section replicated by an active origin ($i+1$ in the figure) reaches origin i before it has had a chance to fire. Replication of $i+1$ continues, overtaking and destroying origin i.

The above discussion suggests that DNA replication is a complex process that involves different types of dynamics: discrete dynamics due to the firing of the origins, continuous dynamics from the evolution of the replication forks, and stochastic terms needed to capture origin efficiencies and uncertainty about their firing times. In the next section we present a stochastic hybrid model to deal with these diverse dynamics.

9.4.3 A PDMP Model

The model splits the genome in pieces whose replication is assumed to be independent of one another. Examples of pieces are chromosomes. Chromosomes may be further divided into smaller pieces, to exclude, for example, rDNA repeats in the middle of a chromosome which are usually excluded in sequencing databases and micro-array data. The model for each piece of the genome requires the following data:

- The length, L, of the piece of the genome, in bases. We will assume that L is large enough so that, if we normalize by L, we can approximate the position along the genome with a continuous quantity, $l \in [0,1]$. This is a reasonable approximation even for the simplest organisms.

- The normalized positions, $O_i \in (0,1)$, $i = 1,2,\ldots,N$, of the origins of replication along the genome. For notational convenience, we append two dummy origins to the list of true origins, situated at the ends of each genome piece, $O_0 = 0$ and $O_{N+1} = 1$.

- The firing rate of the origins, $\lambda_i \in \mathbb{R}_+$, $i = 1,2,\ldots,N$, in $minutes^{-1}$. We set $\lambda_0 = \lambda_{N+1} = 0$.

- The fork velocity, $v(l) \in \mathbb{R}_+$ as a function of the location, $l \in [0,1]$, in the genome.

Using micro-array techniques, values for all these parameters are now becoming available for a number of organisms.

The above discussion reveals that during the S phase, each origin of replication can find itself in one of six discrete states: pre-replicative, *PreR*, replicating on both sides, *RB*, replicating to the right only, *RR*, replicating to the left only, *LR*, post

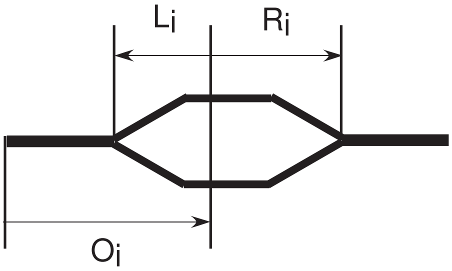

FIGURE 9.7: Definition of continuous states of an origin.

replicating, *PostR*, and passively replicated, *PassR*. The discrete state space of our model will therefore be

$$Q = \{PreR, RB, RR, LR, PostR, PassR\}^N.$$

The discrete state, $q \in Q$ will be denoted as a N-tuple, $q = (q_1, q_2, \ldots, q_N)$ with $q_i \in \{PreR, RB, RR, LR, PostR, PassR\}$. The dummy origins introduced at the beginning and the end of the section of DNA are not reflected in the discrete state, we simply set $q_0 = q_{N+1} = PreR$. Note that the number of discrete states, 6^N, grows exponentially with the number of origins. Even simple organisms have several hundreds of origins and even though only a small fraction of the possible states get visited in any one S phase, the total number of discrete states reached can be enormous.

The number of continuous states depends on the discrete state and will change during the evolution of the system. Since the continuous state reflects the progress of the replication forks, we introduce one continuous state for each origin replicating only to the left, or only to the right and two continuous states for each origin replicating in both directions. Therefore the dimension of the continuous state space for a given discrete state $q \in Q$ will be

$$d(q) = |\{i \mid q_i \in \{RR, LR\}\}| + 2\,|\{i \mid q_i = RB\}|,$$

where, as usual, $|\cdot|$ denotes the cardinality of a set. For an origin with $q_i \in \{RR, RB\}$ we will use R_i to denote the length of DNA it has replicated to its right. Likewise for an origin with $q_i \in \{LR, RB\}$ we will use L_i to denote the length of DNA it has replicated to its left (see Figure 9.7). For a discrete state, $q \in Q$, the continuous state $x \in \mathbb{R}^{d(q)}$ will be a $d(q)$-tuple consisting of the R_i and L_i listed in the order of increasing i; if $q_i = RB$ we assume that the R_i is listed before L_i. Notice that initially all origins will be in the pre-replicative mode and after the completion of the S phase all origins will be in either post replicating or passively replicated. Therefore both at the beginning and at the end of the S phase we will have $d(q) = 0$ and the continuous state space will be trivial.

The open sets $X(q)$ are used to force discrete transitions to take place. Figure 9.6 also summarizes the discrete transitions that can take place for each origin of replication. All transitions except the one from *PreR* to *RB* are forced and have to do with the relation between the replication forks of origin i and those of other replicating origins to its left and to its right. For a discrete state $q \in Q$ and an origin $i = 1, \dots, N$ we denote these replicating neighbors to the left and right of origin i by

$$\begin{aligned} LN_i(q) &= \max\{j < i \mid q_j \in \{RR, RB\}\} \\ RN_i(q) &= \min\{j > i \mid q_j \in \{LR, RB\}\}. \end{aligned}$$

Whenever the sets are empty we set $LN_i(q) = 0$ and $RN_i(q) = N + 1$.

We build the set $X(q)$ out of sets, $X_i(q)$, one of each active origin. Forced transitions occur when replication forks meet. For example, if origin i is only replicating to its right, $q_i = RR$, and its right replication fork, R_i, meets the left replication fork, $L_{RN_i(q)}$, of its right neighbor, $RN_i(q)$, then origin i must stop replicating and switch to $q_i = PostR$. Therefore

$$q_i = RR \Rightarrow X_i(q) = \{O_{RN_i(q)} - L_{RN_i(q)} > O_i + R_i\} \subseteq \mathbb{R}^{d(q)}.$$

Notice that the set is well defined: because $q_i = RR$ and, by definition, $q_{RN_i(q)} \in \{LR, RB\}$ both R_i and $L_{RN_i(q)}$ are included among the continuous states. Likewise, we define

$$\begin{aligned} &q_i = LR \Rightarrow X_i(q) = \{O_{LN_i(q)} + R_{RN_i(q)} < O_i - L_i\} \\ &q_i = RB \Rightarrow X_i(q) = \{O_{LN_i(q)} + R_{RN_i(q)} < O_i - L_i\} \cap \{O_{RN_i(q)} - L_{RN_i(q)} > O_i + R_i\} \\ &q_i = PreR \Rightarrow X_i(q) = \{O_{LN_i(q)} + R_{RN_i(q)} < O_i\} \cap \{O_{RN_i(q)} - L_{RN_i(q)} > O_i\} \\ &q_i \in \{PostR, PassR\} \Rightarrow X_i(q) = \mathbb{R}^{d(q)}. \end{aligned}$$

We define the overall set by

$$X(q) = \bigcap_{i=1}^{N} X_i(q).$$

$X(q)$ is clearly an open set.

The vector field, f, reflects the continuous progress of the replication forks along the genome. It is again defined one origin at a time. We set

$$f_i(q,x) = \begin{cases} v(O_i + R_i) & \in \mathbb{R} \quad \text{if } q_i = RR \\ \begin{bmatrix} v(O_i + R_i) \\ v(O_i - L_i) \end{bmatrix} & \in \mathbb{R}^2 \quad \text{if } q_i = RB \\ v(O_i - L_i) & \in \mathbb{R} \quad \text{if } q_i = LR. \end{cases}$$

Recall that all other discrete states do not give rise to any continuous states. The overall vector field $f(q,x) \in \mathbb{R}^{d(q)}$ is obtained by stacking the $f_i(q,x)$ for the individual replicating origins one on top of the other. Under mild assumptions on the fork velocity it is easy to see that $f(q,\cdot)$ satisfies Assumption 9.1.

The initial state measure is trivial. Biological intuition suggests that at the beginning of the S phase all origins are pre-replicative and no replication forks are active. The initial probability measure is therefore just the Dirac measure

$$Init(q,x) = \delta_{PreR^N \times \{0\}}(q,x).$$

Recall that when $q = PreR^N$ the continuous state is trivial $x \in \mathbb{R}^0 = \{0\}$.

The only spontaneous transition in our model is the one from state *PreR* to the state *RB*; all other transitions are forced. The transition rate, λ, governing spontaneous transitions reflects the randomness in the firing times of the origins. Therefore λ is only important for origins in state *PreR*. We define λ one origin at a time, setting

$$\lambda_i(q,x) = \begin{cases} \lambda_i & \text{if } q_i = PreR \\ 0 & \text{otherwise.} \end{cases}$$

This implies that the firing time, T_i, of origin i have a survival function of the form

$$P[T_i \geq t] = e^{-\lambda_i t}. \tag{9.20}$$

Notice that here T_i refers to the time origin i would fire in the absence of interference from other origins, not the observed firing times. In practice, origin i will sometimes get passively replicated by adjacent origins before it gets a chance to fire. Therefore the observed firing times will show a bias toward smaller values that the $1/\lambda_i$ anticipated by (9.20). We set the overall rate to

$$\lambda(q,x) = \sum_{i=1}^{N} \lambda_i(q,x).$$

Finally, for the transition measure R we distinguish two cases: either no transition is forced (i.e., state before the transition is in $\mathscr{D}$), or a transition is forced (i.e., state before the transition in $\partial\mathscr{D}$). In the former case, for $q \in Q$ let

$$d_i(q) = |\{j < i \mid q_j \in \{RR, LR\}\}| + 2|\{j < i \mid q_j = RB\}|.$$

For $(\hat{q},\hat{x}) \in \mathscr{D}$ with $\hat{q}_i = PreR$ define the measure

$$\delta_{q_i \to RB}(q,x)$$

as the Dirac measure concentrated on $(q,x) \in \mathscr{D}$ with $q_i = RB$, $q_j = \hat{q}_j$ for $j \neq i$, $x_j = \hat{x}_j$ for $j < d_i(\hat{q})$, $x_{d_i(\hat{q})} = x_{d_i(\hat{q})+1} = 0$, and $x_{j+2} = \hat{x}_j$ for $j \geq d_i(\hat{q})$. In words, if origin i fires spontaneously, its discrete state changes to *RB* and two new continuous states are introduced to store the progress of its replication forks. Since a spontaneous transition takes place whenever one of the origins in state *PreR* can fire, the overall reset measure from state $(\hat{q},\hat{x}) \in \mathscr{D}$ can be written as

$$R((q,x),(\hat{q},\hat{x})) = \frac{\sum_{\{i \mid \hat{q}_i = PreR\}} \lambda_i \delta_{q_i \to RB}(q,x)}{\lambda(q,x)}. \tag{9.21}$$

Finally, if $(\hat{q},\hat{x}) \in \partial\mathscr{D}$, i.e., a transition is forced for at least one origin, define the "guard" conditions

$$\begin{aligned}
G_{q_i \to PassR}(\hat{q},\hat{x}) &= (\hat{q}_i = PreR) \wedge \\
&\quad [(O_{LN_i(q)} + R_{RN_i(q)} \geq O_i) \vee (O_{RN_i(q)} - L_{RN_i(q)} \leq O_i)] \\
G_{q_i \to RR}(\hat{q},\hat{x}) &= (\hat{q}_i = RB) \wedge \\
&\quad (O_{LN_i(q)} + R_{RN_i(q)} \geq O_i - L_i) \wedge (O_{RN_i(q)} - L_{RN_i(q)} > O_i + R_i) \\
G_{q_i \to LR}(\hat{q},\hat{x}) &= (\hat{q}_i = RB) \wedge \\
&\quad (O_{RN_i(q)} - L_{RN_i(q)} \leq O_i + R_i) \wedge (O_{LN_i(q)} + R_{RN_i(q)} < O_i - L_i) \\
G_{q_i \to PostR}(\hat{q},\hat{x}) &= [(\hat{q}_i = RB) \wedge \\
&\quad (O_{RN_i(q)} - L_{RN_i(q)} \leq O_i + R_i) \wedge (O_{LN_i(q)} + R_{RN_i(q)} \geq O_i - L_i)] \\
&\quad \vee [(\hat{q}_i = RR) \wedge (O_{RN_i(q)} - L_{RN_i(q)} \leq O_i + R_i)] \\
&\quad \vee [(\hat{q}_i = LR) \wedge (O_{LN_i(q)} + R_{RN_i(q)} \geq O_i - L_i)].
\end{aligned}$$

We can then define $R((\cdot,\cdot),(\hat{q},\hat{x}))$ as a Dirac measure concentrated on (q,x) with

$$q_i = \begin{cases} PassR & \text{if } G_{q_i \to PassR}(\hat{q},\hat{x}) \text{ is true} \\ RR & \text{if } G_{q_i \to RR}(\hat{q},\hat{x}) \text{ is true} \\ LR & \text{if } G_{q_i \to LR}(\hat{q},\hat{x}) \text{ is true} \\ PostR & \text{if } G_{q_i \to PostR}(\hat{q},\hat{x}) \text{ is true} \end{cases}$$

and x same as $\hat{x}$, with the elements corresponding to i with $q_i \neq \hat{q}_i$ dropped. Notice that, as in the case of *B. subtilis*, if forced transitions are available they are taken, preempting any spontaneous transitions.

9.4.4 Implementation in Simulation and Results

The model of DNA replication is very complex, with a potentially enormous number of discrete (6^N) and continuous ($2N$) states. The model has the advantage that it is naturally decomposed to fairly independent components (the models for the individual origins) which interact via their continuous states (the progress of the replication forks). Current research concentrates on exploiting compositional frameworks for stochastic hybrid systems ([2, 28, 32], see also Chapter 3 of this volume) to model and analyze the behavior of the DNA replication mechanism.

In the meantime, the best way to analyze the behavior of this system is through simulation. A simulator of the DNA replication process was developed which simulates the DNA replication process genome wide, given a specific genome size, specific origin positions and efficiencies and specific fork velocities. Event detection was accomplished by computing the zero crossings of functions of the form

$$g(q,x) = O_{RN_i(q)} - L_{RN_i(q)} - O_i - R_i$$

(for the discrete transition from *RR* to *PostR*, and similar functions for the remaining transitions). Servicing the events involved switching the discrete state, but also changing the continuous state dimension, by dropping or adding states.

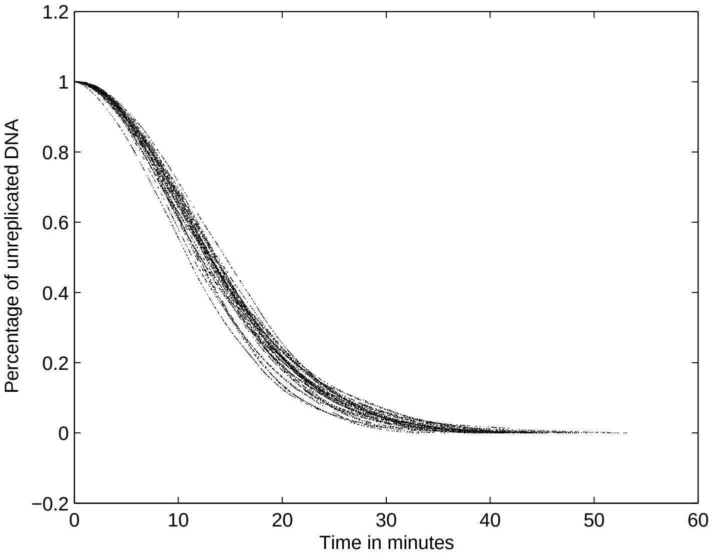

FIGURE 9.8: Evolution of unreplicated DNA.

Simulation results for a number of runs of the model are shown in Figure 9.8. Here the genome size used was 12 million bases, with 900 origins introduced at random locations and with random efficiencies. The fork velocity was constant at 5500 bases per minute. The figure clearly shows the randomness in the DNA replication process predicted by the model.

9.5 Concluding Remarks

We have presented an overview of stochastic hybrid modeling issues that arise in biochemical processes. We have argued that stochastic hybrid dynamics play a crucial role in this context and illustrated this point by developing PDMP models for two biochemical processes, subtilin production by the organism *B. subtilis* and DNA replication in eukaryotes. We also discussed how the models can be analyzed by Monte Carlo simulation. Current research focuses on tuning the parameters of the models based on experimental data and exploiting the analysis and simulation results obtained with the models (in particular the DNA replication model) to gain biological insight. Already the results of the DNA replication model have led biologists to re-think long held conventional opinions about the duration of the S phase and the role of different mechanisms that play a role in cell cycle regulation.

Acknowledgments. The authors are grateful to S. Dimopoulos, G. Ferrari-Trecate, C. Heichinger, H. de Jong, I. Legouras, and P. Nurse for extensive discussions providing insight into the topic. Research supported by the European Commission under the HYGEIA project, FP6-NEST-4995.

References

[1] R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, J. Schug, and G.J. Pappas. Hybrid modeling and simulation of biological systems. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 19–32. Springer-Verlag, Berlin, 2001.

[2] R. Alur, R. Grosu, I. Lee, and O. Sokolsky. Compositional refinement for hierarchical hybrid systems. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 33–48. Springer-Verlag, Berlin, 2001.

[3] K. Amonlirdviman, N. A. Khare, D. R. P. Tree, W.-S. Chen, J. D. Axelrod, and C. J. Tomlin. Mathematical modeling of planar cell polarity to understand domineering nonautonomy. *Science*, 307(5708):423–426, Jan. 2005.

[4] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems*. PhD thesis, Lund Institute of Technology, Lund, Sweden, December 1994.

[5] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, M. Page, and D. Schneider. Qualitative analysis and verification of hybrid models of genetic regulatory networks: Nutritional stress response in *Escherichia coli*. In L. Thiele and M. Morari, editors, *Hybrid Systems: Computation and Control*, number 3414 in LNCS, pages 134–150. Springer-Verlag, Berlin, 2005.

[6] M.L. Bujorianu and J. Lygeros. Reachability questions in piecewise deterministic Markov processes. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, number 2623 in LNCS, pages 126–140. Springer-Verlag, Berlin, 2003.

[7] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Norwell, MA, 1999.

[8] J. Dai, R.Y. Chuang, and T.J. Kelly. DNA replication origins in the *schizosaccharomyces pombe* genome. *PNAS*, 102(102):337–342, January 2005.

[9] M.H.A. Davis. Piecewise-deterministic Markov processes: A general class of

non-diffusion stochastic models. *Journal of the Royal Statistical Society, B*, 46(3):353–388, 1984.

[10] M.H.A. Davis. *Markov Processes and Optimization*. Chapman & Hall, London, 1993.

[11] M.H.A. Davis and M.H. Vellekoop. Permanent health insurance: a case study in piecewise-deterministic Markov modelling. *Mitteilungen der Schweiz. Vereinigung der Versicherungsmathematiker*, 2:177–212, 1995.

[12] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 79:726–739, 2002.

[13] H. de Jong, J.-L. Gouze, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Hybrid modeling and simulation of genetic regulatory networks: A qualitative approach. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, number 2623 in LNCS, pages 267–282. Springer Verlag, Berlin, 2003.

[14] J.F.X. Diffley and K. Labib. The chromosome replication cycle. *Journal of Cell Science* 115:869–872, 2002.

[15] S. Drulhe, G. Ferrari-Trecate, H. de Jong, and A. Viari. Reconstruction of switching thresholds in piecewise-affne models of genetic regulatory networks. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 184–199. Springer Verlag, Berlin, 2006.

[16] E. Fanning M.I. Aladjem. The replicon revisited: an old model learns new tricks in metazoan chromosomes. *EMBO Rep.*, 5(7):686–691, July 2004.

[17] R. Ghosh and C. Tomlin. Lateral inhibition through delta-notch signalling: A piecewise affine hybrid models. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 232–246. Springer-Verlag, Berlin, 2001.

[18] D.M. Gilbert. Making sense of eukaryotic DNA replication origins. *Science*, 294:96–100, October 2001.

[19] J. Hespanha and A. Singh. Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems. *International Journal on Robust Control*, 15(15):669–689, September 2005.

[20] J. Hu, W.C. Wu, and S.S. Sastry. Modeling subtilin production in *bacillus subtilis* using stochastic hybrid systems. In R. Alur and G.J. Pappas, editors, *Hybrid Systems: Computation and Control*, number 2993 in LNCS, pages 417–431. Springer-Verlag, Berlin, 2004.

[21] K.H. Johansson, M. Egerstedt, J. Lygeros, and S.S. Sastry. On the regularization of Zeno hybrid automata. *Systems and Control Letters*, 38(3):141–150, 1999.

[22] M. Kaern, T.C. Elston, W.J. Blake, and J.J. Collins. Stochasticity in gene expression: From theories to phenotypes. *Nature Reviews Genetics*, 6(6):451–464, 2005.

[23] H. Kitano. Looking beyond the details: a rise in system-oriented approaches in genetics and molecular biology. *Curr. Genet.*, 41:1–10, 2002.

[24] J. Lygeros, K.H. Johansson, S.N. Simić, J. Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, January 2003.

[25] J.S. Miller. Decidability and complexity results for timed automata and semilinear hybrid automata. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control*, number 1790 in LNCS, pages 296–309. Springer-Verlag, Berlin, 2000.

[26] P.K. Patel, B. Arcangioli, SP. Baker, A. Bensimon, and N. Rhind. DNA replication origins fire stochastically in the fission yeast. *Molecular Biology of the Cell*, 17(1):308–316, January 2006.

[27] C.V. Rao, D.M. Wolf, and A.P. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420(6912):231–237, 2002.

[28] R. Segala. *Modelling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Techology, 1998.

[29] T. Stein, S. Borchert, P. Kiesau, S. Heinzmann, S. Kloss, M. Helfrich, and K.D. Entian. Dual control of subtilin biosynthesis and immunity in *bacillus subtilis*. *Molecular Microbiology*, 44(2):403–416, 2002.

[30] B. Stern and P. Nurse. A quantitative model for the cdc2 control of S phase and mitosis in fission yeast. *Trends in Genetics*, 12(9), September 1996.

[31] B. Stillman. Origin recognition and the chromosome cycle. *FEBS Lett.*, 579:877–884, February 2005.

[32] S.N. Strubbe. *Compositional Modelling of Stochastic Hybrid Systems*. PhD thesis, Twente University, 2005.

[33] L. Tavernini. Differential Automata and their Simulators. *Nonlinear Analysis, Theory, Methods and Applications*, 11(6):665–683, 1997.

[34] J.J. Tyson, A. Csikasz-Nagy, and B. Novak. The dynamics of cell cycle regulation. *BioEssays*, 24:1095–1109, 2002.

[35] JM. Vilar, HY. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. *PNAS*, 99:5988–5992, 2002.

[36] D. Wolf, V.V. Vazirani, and A.P. Arkin. Diversity in times of adversity: probabilistic strategies in microbial survival games. *Journal of Theoretcial Biology*, 234(2):227–253, 2005.

[37] L. Wu, J.C. Burnett, J.E. Toettcher, A.P. Arkin, and D.V. Schaffer. Stochastic gene expression in a lentiviral positive-feedback loop: HIV-1 tat fluctuations drive phenotypic diversity. *Cell*, 122(2):169–182, 2005.