PARAMETER IDENTIFICATION FOR PIECEWISE DETERMINISTIC MARKOV PROCESSES: A CASE STUDY ON A BIOCHEMICAL NETWORK

Panagiotis Kouretas[†], **Konstantinos Koutoumpas**[†], **and John Lygeros**[†]

[†]*Department of Electrical and Computer Engineering,*
University of Patras, Rio, Patras, GR 26500, Greece

Abstract: The first part of the paper focuses on the use of Piecewise Deterministic Markov Processes (PDMP) for the modeling of biochemical networks. As a case study, the production of the peptide antibiotic *B.Subtilis* is considered. The second part of the paper is devoted to the problem of identifying unkown parameters of the model equations, based on experimental data. For the identification task we propose the use of Genetic Algorithms. An example demonstrates the feasibility of our aproach. *Copyright © 2006 IFAC*

Keywords: Stochastic Hybrid Models, Piecewise Deterministic Markov Processes, Subtilin Production, Genetic Algorithms, Biochemical Networks

1. INTRODUCTION

As far as biology is concerned, the last 20 years marked an astonishing advancement due to pioneering, high-throughput techniques (e.g. micro-arrays) that allowed biologists to shed light to numerous aspects of their field. However researchers realized that the perspective of information gathering was insufficient to provide explanations to difficult questions and lead to the next big step. That possibly explains the recent tendency to express biological processes in terms of a complete system. While in the past, biologists would only observe a target organism, nowadays the first priority is to infer the underlying mechanism that produces the observation.

Such attempts gave birth to a new field called systems biology that applies the aforementioned way of thinking: first a large number of information is collected, then a model for the underlying model is proposed and finally its fitness is validated by a new series of experiments. The general course comprises the thorough observation of the biological process so as to understand all of its characteristics. Then the appropriate model family has to be selected, and finally, each parameter must be refined.

In what follows, we focus our attention on the already studied organism *Bacillus subtilis*. In this paper we emphasize on the conversion of the model that describes the biological process to a valid PDMP model. Piecewise Deterministic Markov Processes are a sub category of stochastic hybrid models (SHM). SHMs were initially developed to capture all the characteristics of systems containing a combination of digital and analogue components and found a wealth of applications in cases such as automated highway systems, air-traffic management systems, manufacturing systems, robotics and real-time communication networks.

Apparently, PDMP seem to be applicable to system biology, apart from being friendly to study and implement. Reasons for choosing this class of SHM will be analyzed later on. Immediately after the expression of the bio-process as a PDMP model, we encounter the difficulty of defining each parameter of the model. The complexity of testing all possible parameter configurations is exponential and would require tremendous computational time and resources. Our work takes into consideration what is biologically feasible and observable in order to identify all the parameter values. Contrary to what is currently present in the literature, our ultimate goal is to infer the parameters and give a valid PDMP model for the target biological process with respect to the quantitative aspect.

2. STOCHASTIC HYBRID MODELS

The great interest of research community for the field of stochastic hybrid systems in recent years led to the introduction of different types of stochastic hybrid models. The main difference between these classes of stochastic hybrid models lies in the way the stochasticity appears [1]. In some models continuous evolution may be governed by stochastic differential equations, while in others not. Likewise, some models include forced transitions, which take place whenever the continuous state tries to leave a given set, others only allow transitions to take place at random times (spontaneously at a given possibly state-dependent rate), while others allow both. Finally the destination of discrete transitions may be given by a probability kernel. In this report the model that will be analyzed is the Piecewise Deterministic Markov Processes [2], which is the class of models that will be used for B-subtilis. Apart from the PDMP there are also the Switched Diffusion Processes (SDP) [3] and the Stochastic Hybrid Systems (SHS) [4]. An overview of them can be found in [1].

2.1 Piecewise Deterministic Markov Processes (PDMP)

PDMPs are a class of non-linear continuous-time stochastic hybrid processes which covers a wide range of non-diffusion phenomena. PDMP involve a hybrid state space, with both continuous and discrete states. The particularity of this model is that randomness appears only in the discrete transitions; between two consecutive transitions the continuous state evolves according to a non-linear ordinary differential equation. 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. We introduce formally PDPM following the notation of ([5], [6]). Let Q be a countable set of discrete states, and let $d : Q \to \mathbb{N}$ and $X : Q \to \mathbb{R}^{d(\cdot)}$ be two maps assigning to each discrete state $i \in Q$ an open subset of $\mathbb{R}^{d(i)}$. We call the set

$$\mathcal{D}(Q,d,X) = \bigcup_{i \in Q} \{i\} \times X(i) = \{(i,x) : i \in Q, x \in X(i)\}$$

the hybrid state space of the PDMP and $\alpha = (i,x) \in \mathcal{D}(Q,d,X)$ the hybrid state. We define the boundary of the hybrid state space as

$$\partial\mathcal{D}(Q,d,X) = \bigcup_{i \in Q} \{i\} \times \partial X(i).$$

where as usual $\partial X(i)$ denotes the boundary of the open set $X(i)$.

A vector field f on the hybrid state space $\mathcal{D}(Q,d,X)$ is a function $f : \mathcal{D}(Q,d,X) \to \mathbb{R}^{d(\cdot)}$ assigning to each hybrid state $(i,x) \in \mathcal{D}$ a direction $f(i,x) \in \mathbb{R}^{d(i)}$. The flow of f is a function $\Phi : \mathcal{D}(Q,d,X) \times \mathbb{R} \to \mathcal{D}(Q,d,X)$ with

$$\Phi(i,x,t) = \begin{bmatrix} \Phi_Q(i,x,t) \\ \Phi_X(i,x,t) \end{bmatrix},$$

$\Phi_Q(i,x,t) \in Q$ and $\Phi_X(i,x,t) \in X(i)$, such that (i,x), $\Phi(i,x,0) = i,x$ and for all $t \in \mathbb{R}$, $\Phi_Q(i,x,t) = i$ and

$$\frac{d}{dt}\Phi_X(i,x,t) = f(\Phi(i,x,t)) \qquad (2.1)$$

Let

$$\Gamma((Q,d,X),f) =$$

$$\{\alpha \in \partial\mathcal{D}(Q,d,X) \mid \exists(\alpha',t) \in \mathcal{D}(Q,d,X) \times \mathbb{R}^+, \alpha = \Phi(\alpha',t)\}$$

Denotes the part of the boundary of $\mathcal{D}$ that can be reached from $\mathcal{D}$ under f and let

$$\bar{\mathcal{D}}(Q,d,X) = \mathcal{D}(Q,d,X) \cup \Gamma((Q,d,X),f)$$

Let $\bar{\mathcal{D}} = Q \times \mathbb{R}^\infty$ and

$$\mathcal{B}(\bar{\mathcal{D}}) = \sigma\left(\bigcup_{i \in Q} \{i\} \times \mathcal{B}(i)\right)$$

be the smallest σ - algebra on $(\bar{\mathcal{D}}$ containing all sets of the form $i \times A$ with $A \in B(i)$ a Borel subset of $X(i)$. It can be shown that the space $(\bar{\mathcal{D}}, \mathcal{B}(\bar{\mathcal{D}}))$ is a Borel Space and $\mathcal{B}(\bar{\mathcal{D}})$ is a sub-σ-algebra of its Borel σ-algebra.

We can now introduce the following definition.

Definition 1. (**Piecewise Deterministic Markov Process**). A Piecewise Deterministic Markov Process (PDMP) is a collection $H = ((Q,d,X), f, Init, \lambda, R)$ where

- Q is a countable set of discrete variables;
- $d : Q \to \mathbb{N}$ is a map giving the dimensions of the continuous state spaces;
- $X : Q \to \mathbb{R}^{d(\cdot)}$ maps each $i \in Q$ into an open subset $X(i)$ of $\mathbb{R}^{d(i)}$;
- $f : \mathcal{D}(Q,d,X) \to \mathbb{R}^{d(\cdot)}$ is a vector field;
- $Init : \mathcal{B}(\bar{\mathcal{D}}) \to [0,1]$ is an initial probability measure on $(\bar{\mathcal{D}}, \mathcal{B}(\bar{\mathcal{D}}))$, with $Init(\mathcal{D}^c) = 0$;
- $\lambda : \bar{\mathcal{D}}(Q,d,X) \to \mathbb{R}^+$ is a transition rate function;
- $R : \mathcal{B}(\bar{\mathcal{D}}) \times \bar{\mathcal{D}}(Q,d,X) \to [0,1]$ is a transition measure, with $R(\mathcal{D}^c,(i,x)) = 0$ for all $(i,x) \in (\bar{\mathcal{D}}(Q,d,x)$.

To define the PDMP executions we introduce the notion of exit time $t^* : \mathcal{D} \to \mathbb{R}^+ \cup \{\infty\}$,

$$t^*(i,x) = \inf\{t > 0 : \Phi(i,x,t) \notin \mathcal{D}\}$$

and of survival function $F : \mathcal{D} \times \mathbb{R}^+ \to [0,1]$,

$$F(i,x,t) = \begin{cases} \exp\left(-\int_0^t \lambda(\Phi(i,x,\tau))d\tau\right) & \text{if } t < t^*(i,x) \\ 0 & \text{if } t \geq t^*(i,x). \end{cases}$$

The executions of the PDMP can be thought of as being generated by the following algorithm.

Algorithm 1. (**Generation of PDMP Executions**)

```
    set $T = 0$
    select $\mathcal{D}$-valued random variable $(\hat{i}, \hat{x})$ according to $Init$
    repeat
        select $\mathbb{R}^+$-valued random variable $\hat{T}$ such that $P(\hat{T} > t) = F(\hat{i}, \hat{x}, t)$
        set $(i_t, x_t) = \Phi(\hat{i}, \hat{x}, t - T)$ for all $t \in [T, T + \hat{T})$
        select $\mathcal{D}$-valued random variable $(\hat{i}, \hat{x})$ according to $R(., \Phi(\hat{i}, \hat{x}, \hat{T}))$
        set $T = T + \hat{T}$
    until true
```

To ensure the process is well-defined, the following assumption is introduced in [2].

Assumption 1. The sets $X(i)$ are open. For all $i \in Q$, $f(i, .)$ is globally Lipschitz continuous. $\lambda : \bar{\mathcal{D}}(Q, d, X) \to \mathbb{R}^+$ is measurable. For all $i, x \in \mathcal{D}$ there exists $\varepsilon > 0$ such that the function $t \to \lambda(\Phi(i, x, t))$ is integrable for all $t \in [0, \varepsilon)$. For all $A \in \mathcal{B}(\bar{\mathcal{D}})$, $R(A, \cdot)$ is measurable.

All random extractions in Algorithm 1 are assumed to be independent. To ensure that (i_t, x_t) is defined on the entire $\mathbb{R}^+$ it is necessary to exclude Zeno executions [6]. The following assumption is introduced in [2] to accomplish this.

Assumption 2. Let $N_t = \sum_i I_{(t \geq T_i)}$ be the number of jumps in $[0, t]$. Then $\mathbb{E}[N_t] < \infty$ for all t.

Under Assumptions 1, 2 it can be shown that the Algorithm 1 defines a strong Markov process [2], continuous from the right with left limits.

3. MODEL OF SUBTILIN PRODUCTION

In order to display the descriptive power of PDMP, we develop a model for the system that governs Subtilin production by *B. subtilis* bacterium in terms of Piecewise Deterministic Markov Processes(PDMP). This form of stochastic hybrid model was selected because it coincides the inherent characteristics of the model. In the following sections, we will focus on special characteristics, and we will provide a consummate PDMP description.

3.1 Subtilin production

Subtilin is an antibiotic released by *B. subtilis* as a way to confront 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 in general). 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 [8]. *B. subtilis* 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 [7], subtilin derives from the peptide SpaS. Responsible for the production of SpaS is the activated protein SpaRK, which is composed in turn by the binding of the SigH protein to upstream genes of SpaS protein. Finally, the composition of SigH is turned on whenever the nutrient concentration falls below a certain threshold.

3.2 Model equations

A stochastic hybrid model for this process was proposed in [7]. The equations of the model were developed based on the qualitative description of Section 3.1. 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 [7] 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 SpaS at the time $k\Delta$, $x_3(k\Delta)$. More specifically, this probability is

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

where c is a model constant. 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

$$a_1(x_3) = \frac{1}{1 + cx_3}.$$

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.

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

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

Under this equation, x_1 will tend to converge to D_∞, the steady state population for a given nutrient amount. D_∞ depends on x_2 and is given by

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

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 continuous dynamics for x_2 are governed by:

$$\dot{x}_2 = -k_1x_1 + k_2x_3$$

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. In reality, the second term is proportional to the average concentration of SpaS, but for simplicity we follow [7] and assume that the average concentration is proportional to the concentration of SpaS for a single cell.

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

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

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 Piecewise Affine (PWA) with the exeption of nonlinear x_1 and the stochastic terms used to describe switch behavior.

3.3 PDMP formalism

We now try to express the model for subtilin production using the PDMP formalism. As presented earlier, a Piecewise Deterministic Markov Process is a collection $H = ((Q, d, X), f, Init, \lambda, R)$. We saw that the subtilin production has 8 discrete states. Therefore, the set Q is a countable set, the cardinality of which is 8. Let

$$Q = q_0, ...q_7,$$

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 swithes 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 $q_0, ...q_7$ 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 $d(q) = 5$, for all $q \in Q$. The definition of the survival function 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 subtilin 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

$$X(0**) = \mathbb{R} \times (\eta, \infty) \times \mathbb{R}^3$$

and

$$X(1**) = \mathbb{R} \times (-\infty, \eta) \times \mathbb{R}^3$$

The above elements completely determine the hybrid state space, $\mathcal{D}(Q, d, X)$, of the PDMP. As far as the vector field is examined, we have to specify the direction $f(a)$ that is assigned to each state $a = (q, x)$. The vector field is not dependent on the value of X but depends on the discrete states. Therefore, we have:

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

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

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

$$f(q_6, x) = \begin{bmatrix} rx_1(1 - \frac{x_1}{D_\infty}) \\ -k_1x_1 + k_2x_3 \\ k_3 - l_3x_3 \\ k_4 - l_4x_4 \\ -l_5x_5 \end{bmatrix} \quad f(q_7, x) = \begin{bmatrix} rx_1(1 - \frac{x_1}{D_\infty}) \\ -k_1x_1 + k_2x_3 \\ k_3 - l_3x_3 \\ k_4 - l_4x_4 \\ k_5 - l_5x_i \end{bmatrix} \tag{3.1}$$

Regarding the initial state of the model, for simplicity reasons, as well as for biological common sense, we assume that executions start always from the q_0 discrete state. Also, we require that the probability distribution Init satisfies

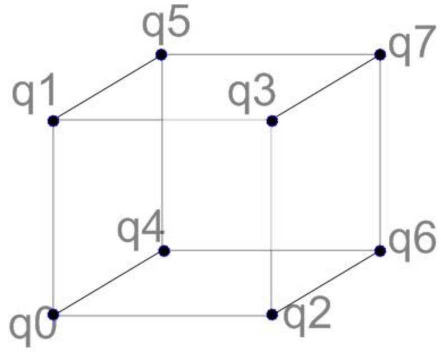

Fig. 1. discrete state space

$Init(0 ** \times x \in \mathbb{R}^5 | x_2 \leq \eta) = 0,$
$Init(1 ** \times x \in \mathbb{R}^5 | x_2 \geq \eta) = 0.$

The initial state should reflect any other constraints imposed by biological intuition. For example, since x_1 reflects the *B.Subtilis* population, it is reasonable to assume that they $x_1 \geq 0$. Another reasonable constraint is that initially $x_1 \leq D_\infty(x_2)$. Finally, since continuous states $x_2, ..., x_5$ 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 $q \in Q$

$$Init(\{q\} \times \{x \in \mathbb{R}^5 | \; x_1 \in (0, D_\infty(x_2)) \text{ and } min(x_2, x_3, x_4, x_5) > 0\}) = 1 \tag{3.2}$$

All probability distributions that respect the above constraints are considered acceptable for our model.

The main problem we confront when trying to express the subtilin production model as a PDMP is the need to define the λ function. Intuitively, this function indicates the "tendency" of the system to jump and switch its discrete state. The rate function λ will govern the spontaneous transitions of the switches S_4 and S_5 (switch S_3 is governed by a forced transition). To present the design of an appropriate λ function we focus on discrete state q_6. Figure 2 summarizes the discrete transitions out of state q_6. Simultaneous switching of more than one of the switches S_3, S_4, S_5 is not allowed. This is a reasonable assumption, since simultaneous switching of two or more switches is a null event in the unerlying probability space. 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. For the remaining two transitions, we define two separate rate functions, $\lambda_{S_4 \rightarrow OFF}(x)$ and $\lambda_{S_5 \rightarrow ON}(x)$. 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 [7]. The survival function of states that the probability that the switch S_4 remains ON throughout the interval $[(k-1)\Delta, k\Delta]$ is equal to

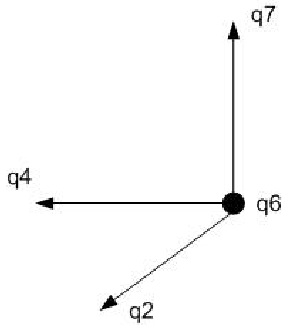

Fig. 2. possible transitions

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

This propability should be equal to $1 - a_0(x_3((k-1)\Delta))$. Assuming that Δ is small enough, we have that $1 - a_0(x_3((k-1)\Delta)) \approx exp\left(-\Delta\lambda_{S_4 \rightarrow OFF}(x(k\Delta))\right)$. Selecting $\lambda_{S_4 \rightarrow OFF}(x) = \frac{ln(1+cx_3)}{\Delta}$ achieves the desired effect. Likewise, we define $\lambda_{S_5 \rightarrow ON}(x) = \frac{ln(1+cx_4) - ln(cx_4)}{\Delta}$ and set the transition rate for discrete state q_6 to $\lambda(q6, x) = \lambda_{S_4 \rightarrow OFF}(x) + \lambda_{S_5 \rightarrow ON}(x)$

The functions $\lambda_{S_4 \rightarrow OFF}(x)$ and $\lambda_{S_5 \rightarrow ON}(x)$ take non-negative values and are therefore good candidates for rate functions. In a similar way, we define rate functions $\lambda_{S_5 \rightarrow OFF}(x)$ (replacing x_3 by x_4) and $\lambda_{S_4 \rightarrow ON}(x)$ (replacing x_4 by x_3) and use them to define the transition rates for the remaining discrete states. In order to complete the PDMP model we need to define 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. We do this by introducing the a priority scheme: when 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

$$R(q_6, x) = \delta_{(q_2, x)}(q, x) \text{ if } (q_6, x) \in \mathcal{D} \text{ else}$$
$$R(q_6, x) = \frac{\lambda_{S_4 \rightarrow OFF}(x)}{\lambda(q6, x)}\delta_{(q_4, x)}(q, x) + \frac{\lambda_{S_5 \rightarrow ON}(x)}{\lambda(q6, x)}\delta_{(q_7, x)}(q, x) \tag{3.3}$$

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 can be written together using the indicator function, $I_{\mathcal{D}}(q, x)$, of the set $\mathcal{D}$. It is easy to see that this probability measure satisfies Assumpion 1.

The above discussion shows that the PDMP model also satisfies most of the conditions of Assumption 1. The only problem may be the non-Zeno condition.

4. PARAMETER IDENTIFICATION

4.1 Problem formulation

In this section we will concentrate on formulating mathematically the problem, in order to make it solvable

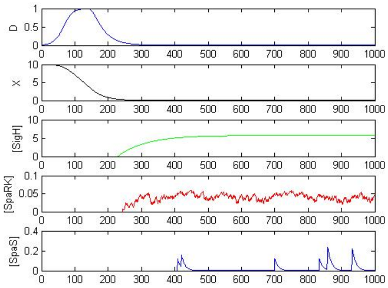

Fig. 3. Execution for randomly selected values

by the use of Genetic Algorithms. First of all, we must clarify that our intention is to estimate only the parameters involved in the differential equations governing the function $f(x)$ of the PDMP formalism. We have to emphasize at this point, that only the first two curves of the Fig 3 (showing the evolution of x_1 and x_2) are to be used, because they comprise the only observable data of the system. The other three curves indicate the evolution of intracellular concentrations, and are not readily available for measurement. Therefore, the problem can be expressed as: Is it possible to exploit the curves expressing the food and the population evolution in order to reveal the values of the parameters of the underlying model? In what follows, we present our effort to estimate the values by measuring the distance of a generated curve (based on random selection of parameters) from its target-curve (the original one). That is, we treat these curves as bearing all the information hidden in the system in the form of parameter values.

Figure 3 shows the execution of the five continuous states of the aforementioned system in case of random selection of the target values, so as to become evident that the a priori knowledge of the structure of the system is not sufficient to guarantee reasonable results.

The exact set of parameters we want to identify are the five values of synthesis rates k_1 to k_5, three degradation rates l_3 to l_5, the constant r, the time interval δ and the threshold η.

4.2 Proposed solution based on GA

The genetic algorithm is a method for solving optimization problems. They are based on natural selection, and are inspired by the Darwinian optimization process that governs evolution in real life. The genetic algorithm first creates and then modifies a set of individual solutions. At each step, the genetic algorithm must select a subset of individuals from the given population for mating reasons. The selected individuals produce the population of the next generation. Over successive generations, we expect the population to evolve toward a better solution, according to a fitness function. Researchers have proposed numerous slight modifications (concerning a GA's evolution). However, the three main types of rules of a GA are the Selection rules, the Crossover rules and the Mutation rules. The next paragraph describes implementation topics of our genetic algorithm application.

The simulation was held on the MATLAB environment for ease of use [9]. We exploited the conveniences provided by the version 1.0.1 of the Genetic Algorithm toolbox to obtain our results. In order to run the algorithm, we must firstly designate all the parameters of the algorithm informing properly all the fields of the structure *gaoptimset*.

CreationFcn: our selection is the function *gacreationuniform* for we want to initialize a uniform population

CrossoverFraction: denotes the fraction of the population to be created from one generation to the next by the crossover. It is set to the default value 0.8.

CrossoverFcn: informs the GA in what way to create crossover children. We selected *crossoverscattered*

PopInitRange: Vector needed to inform of the boundary values. In our case it coincides with the default value [0;1] because the majority of the parameters must operate inside this area.

PopulationType: The data type of the population could be 'bitstring','doubleVector', or customized. The first two do not serve our purposes, consequently we had to use the 'custom' option because our values must be positive double values. Besides that, we had to render the selected creation and mutation functions operational with our data type, including changes in the source code of the respective functions

MutationFcn: Governs the way that mutations are generated. *Gaussian* mutation was selected in order to produce generally small perturbations.

PopulationSize: This number shows the number of the population. The greater the number, the more precise our search of the state space (but also the more slow the search becomes).

SelectionFcn: The function that selects individuals for mating purposes is set to the commonly used *selectionroulette*. This is a selection operator in which the chance of an individual getting selected is proportional to its fitness. This is where the concept of survival of the fittest comes into play.

The **fitness function** is perhaps the most important parameter that we can define. It was noted earlier that the trajectories of the food as well as the population evolution have to be considered as evaluation means. If we simply calculate the mean square distance between the original trajectories and those generated by a different set of parameters, we do not take into consideration the stochastic nature of the model. The results include different sets of parameters that reproduce satisfactorily the observed trajectories.

The following formula describes our fitness function. That is, for every set of parameters we calculate the evolution of x_1 and x_2. Then these curves are sampled with sampling interval δ. Small letters are used to denote the generated curves, while capital letters denote the target curves.

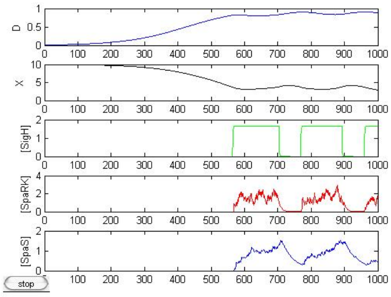

Fig. 4. Results generated by simple fitness function

$$\sum_k [(x_1(k\delta) - X_1(k\delta))^2 + (x_2(k\delta) - X_2(k\delta))^2]$$

Consequently, the previous formula measures the square distance,at specific time points of the evolution, between the target curves (which have to be observed experimentally) and the original ones. Results are shown in Figure 4.

4.3 Results for B.Subtilis model

The PDMP model for Subtilin production comprises randomness that clearly leads each execution to differ from every other (same set of parameters and initial conditions). Thus, we believe that a sole execution is not adequate to fully characterize a given set of values.

Our intention is then to run multiple experiments in order to calculate multiple measurements of mean square distances between the original and the under-examination trajectories. The mean arithmetic value serves then as a fitness indicator.

$$\sum_{i=1}^{2} \frac{\sum_k [(x_1^i(k\delta) - X_1(k\delta))^2 + (x_2^i(k\delta) - X_2(k\delta))^2]}{2}$$

We now extend the fitness function shown before, in order to capture more than one experiments. More specifically, we simulate the evolution that is forced by a set of parameters and then calculate the simple square distance. Then we re-simulate the evolution of the system, and calculate its distance once more. Due to stochastic nature of the system, its evolution (and therefore its distance from the target curves) will differ from execution to execution. In our example $i = 2$, showing that two executions of the system are taken into consideration. Generated results are shown in Figure 5

4.4 Conclusion

PDMPs are a class of models that capture inherent randomness. Their stochastic nature makes it difficult to apply parameter identification techniques to find the parameter values of the $f(x)$ function. However, genetic algorithms were able, only by measuring the mean square distance of the population and food trajectories, to estimate a fair set of parameters, that produce executions, matching the original curves.

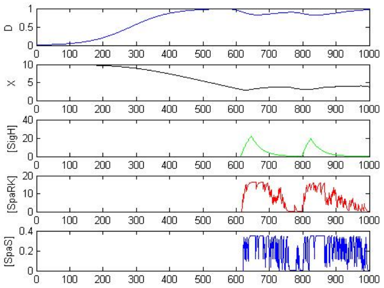

Fig. 5. Results generated by "multiple" fitness function

The fact that an increasing number of researchers try to describe biological processes in terms of stochastic hybrid models, along with the friendly nature of the PDMP formalism, makes us believe that the latter will be further used in the near future in the field of systems biology. We expect genetic algorithms to continue to serve as a means of parameter identification technique for such cases.

5. ACKNOWLEDGMENT

Research supported by the European Commission under the project HYGEIA, NEST-4995. The authors wish to thank the anonymous referees for their careful reading of the manuscript and their fruitful comments.

REFERENCES

1. G. Pola, M. Bujorianu, J. Lygeros, and M. Benedetto, Stochastic hybrid models: An overview, Science,vol. 2,.
2. M. Davis, Markov Processes and Optimization. London: Chapman and Hall, (1993)
3. M. Ghosh, A. Arapostathis, and S. Marcus, Ergodic control of switching diffusions, SIAM Journal on Control Optimization, vol. 35, pp. 19521988.
4. J. Hu, J. Lygeros, and S. Sastry, Towards a theory of stochastic hybrid systems, no. 1790, pp. 160173, 2000.
5. M. Bujorianu and J. Lygeros, Reachability questions in piecewise deterministic markov processes, in Submitted to HSCC.
6. J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. Sastry, Dynamical properties of hybrid automata, IEEE Transactions on Automatic Control, 2003.
7. J. Hu, W. Wu, and S. Sastry, Modeling subtilin production in bacillus subtilis using stochastic hybrid systems, The Seventh International Workshop on Hybrid Systems: Computation and Control, 2004.
8. T. Stein and S. Borchert and P. Kiesau and S. Heinzmann and S. Kloss and M. Helfrich and K.D. Entian, *Dual Control of Subtilin Biosynthesis and Immunity in Bacillus Subtilis*, Molecular Microbiology. Vol. 44, 2002, 403-416.
9. www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml