RANDOMIZED OPTIMIZATION METHODS IN PARAMETER IDENTIFICATION FOR BIOCHEMICAL NETWORK MODELS

K. Koutroumpas [*,1], **E. Cinquemani** [*] **and J. Lygeros** [*]

[*] *Automatic Control Laboratory, ETH Zürich*
Physikstrasse 3, 8092 Zürich, Switzerland

Abstract: The application of randomized optimization techniques to the parameter identification of a model of subtilin production by *Bacillus subtilis* is discussed. A Markov chain Monte Carlo approach to identification is presented along with an initialization method based on genetic algorithms. The combination of the two optimization techniques is shown to improve Markov chain Monte Carlo estimation performance while preserving theoretical convergence properties that are not offered by the genetic algorithms alone. Results from numerical simulations are reported.

Keywords: Stochastic Hybrid Models, Parameter Identification, Subtilin Production, Genetic Algorithms, Markov Chain Monte Carlo

1. INTRODUCTION

Biological processes exhibit dynamics that range over a very wide time scale and contain stochastic components and often discrete components as well. 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 (Ghosh and Tomlin, 2001; Alur *et al.*, 2001; de Jong *et al.*, 2003; Batt *et al.*, 2005; Amonlirdviman *et al.*, 2005; Drulhe *et al.*, 2006).

In this paper we investigate the problem of identification of a model of subtilin production by *Bacillus subtilis*. We shall rely on the stochastic hybrid model introduced in (Hu *et al.*, 2004) and discuss the estimation of the model parameters from partial measurements of the state. This problem was first investigated in (Koutroumpas *et al.*, 2006), where a randomized approach to identification was considered. The problem was formulated as the optimization of a suitable function of fit between experimental data and synthetic data generated by the model for changing values of the parameters. A Genetic Algorithm (GA) was implemented to solve optimization numerically. Results indicate that GAs perform nicely and are little sensitive to the initial parameter guesses. However, several different estimates appeared to explain the data equally well, pointing out a (possible) issue of (weak) parameter identifiability. To gain better insight into the problem, the same problem is tackled here by Markov Chain Monte Carlo (MCMC) techniques. Contrary to GAs, these iterative stochastic approximation procedures are supported by theoretical convergence properties. In addition, MCMC provides not just isolated parameter estimates, but a family of parameter estimates that reflects the structure of the identification problem at hand. In our experience, convergence of MCMC is critically dependent on the initial parameter space search policy. Therefore, a further refinement of the method is proposed in which GAs are used to properly

[1] Corresponding author: koutroumpas@control.ee.ethz.ch.

initialize MCMC. In analogy with (Koutroumpas *et al.*, 2006), the study of identification is carried out on synthetic experimental data generated by simulating the model with suitable parameter values. This simplifies the problem in that the model structure is assumed to be correct, and is mainly due to true experimental data being currently unavailable to us. On the other hand, it allows us a straightforward evaluation of the identification methods.

The present work is organized as follows. Section 2 describes the stochastic hybrid model of subtilin production. The identification problem is formulated in Section 3. Subsections 3.1 and 3.2 discuss the optimization techniques that will be used for identification. Section 4 presents the identification results obtained by applying MCMC and by coupling GA and MCMC. Conclusions and future objectives are discussed in Section 5.

2. SUBTILIN PRODUCTION

Subtilin is an antibiotic synthesized by bacteria *B.subtilis* as an adaptive response to changes in the environment. Subtilin production starts when the amount of nutrient falls under a threshold because of excessive population growth. The role of subtilin is to increase food supply by eliminating competing species and weaker *B.subtilis* cells. In addition to reducing the demand for nutrients, the decomposition of the cells killed by subtilin releases additional nutrients in the environment. The biosynthesis of subtilin is regulated by a positive feedback mechanism in which extracellular subtilin activates the two-component regulatory system SpaK and SpaR that binds to a DNA motif promoting the expression of genes for subtilin synthesis (spaS and spaBTC) and immunity (spaIFEG). SpaK and SpaR react to form the complex SpaRK that will be used in our model. SpaRK expression is controlled by the sporulation transcription factor SigH. Finally, the composition of SigH is turned on whenever the nutrient concentration falls below a certain threshold. In this paper a simplified model is examined, in which spaBTC and spaIFEG are not taken into consideration.

2.1 Stochastic Hybrid Model

A detailed description of a stochastic hybrid model for subtilin production is provided in (Hu *et al.*, 2004). The normalized population of *B.subtilis*, the amount of nutrient and the concentrations of SigH, SpaRK and SpaS constitute the five continuous states $(x_1, x_2, x_3, x_4, x_5)$ of the model. In addition, three binary switches (S_3, S_4 and S_5), give rise to $2^3 = 8$ discrete states. 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 above this threshold. The other two switches are stochastic. In (Hu *et al.*, 2004) this stochastic behavior is approximated by a discrete-time Markov chain, with constant sampling interval Δ. Given that 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

$$a_0(x_3) = cx_3/(1+cx_3). \tag{1}$$

Notice that the probability of switching ON increases to 1 as x_3 gets higher. Conversely, given that S_4 is ON at time $k\Delta$, the probability that it will be OFF at time $(k+1)\Delta$ is

$$a_1(x_3) = 1 - a_0(x_3) = 1/(1+cx_3). \tag{2}$$

This probability increases to 1 as x_3 gets smaller. The dynamics of S_5 are similar, with the concentration of SpaRK, x_4, replacing x_3 and a different value, c', for the constant. The growth of *B.subtilis* population (x_1) is given by the logistic equation

$$\dot{x}_1 = rx_1\big(1 - x_1/D_\infty(x_2)\big). \tag{3}$$

Under this equation, x_1 will tend to converge to

$$D_\infty(x_2) = \min\left\{x_2/X_0, D_{\max}\right\}, \tag{4}$$

the steady state population for a given nutrient amount. X_0 and $D_{\max}$ are constants of the model. The dynamics for x_2 are governed by

$$\dot{x}_2 = -k_1x_1 + k_2x_5 \tag{5}$$

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 (Hu *et al.*, 2004) assume that the average concentration is proportional to the concentration of SpaS for a single cell. Even though this approximation may be inaccurate in some cases we will not attempt to refine the model further here. The dynamics for the remaining three states depend on the discrete state, i.e., the state of the three switches. In all three cases,

$$\dot{x}_i = \begin{cases} -l_ix_i & \text{if } S_i \text{ is OFF} \\ k_i - l_ix_i & \text{if } S_i \text{ is ON.} \end{cases} \tag{6}$$

k_3, k_4 and k_5 represent the synthesis rates of SigH, SpaRK and SpaS when the corresponding genes are ON; l_3, l_4 and l_5 represent the natural decay rates. The model can be formulated in the context of piecewise deterministic Markov processes (Davis, 1984) and can be easily coded in simulation, see (Kouretas *et al.*, 2006) for details. A sample run of the model is given in Figure 1. Notice that, for the given initial conditions, the early evolution of the system is deterministic; stochastic phenomena do not appear before the seventh

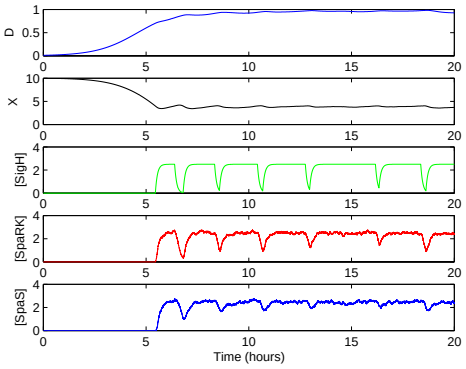

Fig. 1. One execution of the true model.

hour, when the subtilin production mechanism is triggered for the first time.

3. IDENTIFICATION

We assume that the "experimental" data consist of noisy observations of regularly sampled food and population profiles. As was mentioned in the introduction, experimental data are replaced by synthetic data generated by the model:

$$Y_{\theta^*} = [Y_1(\delta|\theta^*)\ Y_2(\delta|\theta^*) \dots Y_1(n\delta|\theta^*)\ Y_2(n\delta|\theta^*)]^T$$

where $Y_1(k\delta|\theta^*)$ and $Y_2(k\delta|\theta^*)$ denote nutrient level and population measurements, in the order, at the observation times $k\delta$, with $k = 1, \dots, n$, and θ^* is the vector of "true" parameter values. Given data Y_{θ^*}, we wish to estimate the following vector of parameters:

$$\theta = (k_2, k_3, k_4, k_5, l_3, l_4, l_5).$$

The remaining parameters of the model, namely r, k_1, η, c and c', are assumed to be known. These parameters could be inferred from the early deterministic evolution of x_1 and x_2 with little difficulty, under the biologically relevant assumption that the initial food level is above threshold η and that $x_3(0) = x_4(0) = x_5(0) = 0$ (i.e. stochastic subtilin production is initially inhibited). Let $J(\theta|Y_{\theta^*})$ be a fitness function that quantifies how well the model with parameters θ explains the data Y_{θ^*}; the larger the value, the better the fit. The identification problem is formalized as follows: find

$$\hat{\theta} \in \arg\sup_{\theta \in \Xi} J(\theta|Y_{\theta^*}), \tag{7}$$

where Ξ is the space of admissible parameter values. Given the complexity of the subtilin production model, we choose to evaluate candidate parameter vectors θ by simulation. That is, nutrient level and population data

$$Y_\theta = [Y_1(\delta|\theta)\ Y_2(\delta|\theta) \dots Y_1(n\delta|\theta)\ Y_2(n\delta|\theta)]^T.$$

are synthetically generated by running the model with parameter θ, and are compared to Y_{θ^*} by way of $J(\theta|Y_{\theta^*})$. In practice, given the stochastic nature of the system, several outcomes of Y_θ and Y_{θ^*} will be considered. These will be indicated by $Y^i_{\theta^*}$, with $i = 1, \dots, N$, and Y^j_θ, $j = 1, \dots, M$, and will be assumed to be statistically independent of each other. As a fitness function we shall use the Normalized Cross Correlation (NCC)

$$J(\theta|Y_{\theta^*}) \triangleq \mathbb{E}\left[\frac{{Y_{\theta^*}}^T Y_\theta}{\sqrt{({Y_{\theta^*}}^T Y_{\theta^*})({Y_\theta}^T Y_\theta)}}\right] = \mathbb{E}\left[\frac{Y_{\theta^*}}{\sqrt{{Y_{\theta^*}}^T Y_{\theta^*}}}\right]^T \mathbb{E}\left[\frac{Y_\theta}{\sqrt{{Y_\theta}^T Y_\theta}}\right] \tag{8}$$

where the expectations are computed empirically from the data by the formula

$$\frac{1}{N}\frac{1}{M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\frac{{Y^i_{\theta^*}}^T Y^j_\theta}{\sqrt{({Y^i_{\theta^*}}^T Y^i_{\theta^*})({Y^j_\theta}^T Y^j_\theta)}}\right). \tag{9}$$

This corresponds to aligning the mean of the simulated data (i.e. of the estimated model) to that of the experimental data while favoring small variance. As shown in (Koutroumpas *et al.*, 2006) by model validation, NCC outperforms other relevant fitness functions.

3.1 Genetic Algorithm

Genetic Algorithms (GA) are stochastic search methods inspired by the evolutionary ideas of natural selection. The basic concept of GA is to mimic the mechanisms of natural selection that follow the principles of "survival of the fittest". Whereas most stochastic search methods operate on a single candidate solution of the problem at hand, genetic algorithms operate on a population of solutions (of constant size). In the language of GAs, a "genome" is the binary coding of a candidate solution for an unknown vector of parameters θ. The GAs use various selection criteria to pick the individuals with the best genome for "mating". A fitness function (in this work, NCC) determines how "good" each individual is and determines its mating probability. After the mating individuals have been selected, suitable stochastic "crossover" and "mutation" rules are applied to them to create the individuals of the next generation (i.e. the new parameter guesses). Randomness ensures that the algorithm keeps exploring the parameter space and alleviates the risk of getting stuck into local maxima. In (Koutroumpas *et al.*, 2006) it has been shown that a suitable implementation of GA provides approximate solutions of (7) that explain the observed data reasonably well, but poor parameter estimation accuracy is attained. This is possibly due to the small sensitivity of the nutrients and population outcomes to changes in the unknown parameter θ, and it is unclear

whether improvements may be obtained by repeated runs of genetic optimization. For the sake of conciseness, the details of the implementation are omitted here. The interested reader is referred to (Koutroumpas *et al.*, 2006).

3.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are iterative algorithms for sampling from a probability distribution that is too complex to be handled directly. The idea is to construct a Markov chain that has the desired distribution as its stationary distribution. Literature provides many variants of MCMC, all of them relating to the general framework of Metropolis (Metropolis *et al.*, 1953) and Hastings (Hastings, 1970). In the *Metropolis–Hastings* algorithm, at each iteration t, the next state Z_{t+1} of the chain is chosen by drawing a candidate point $\tilde{Z}$ from a convenient proposal distribution $q(\cdot|Z_t)$. The candidate point $\tilde{Z}$ is then accepted with probability $\alpha(Z_t, \tilde{Z})$, where

$$\alpha(z, \tilde{z}) = \min\left(1, \pi(\tilde{z})q(z|\tilde{z})/\pi(z)q(\tilde{z}|z)\right)$$

where π is the desired stationary distribution. If the candidate point is accepted, the next state becomes $Z_{t+1} = \tilde{Z}$, else the chain does not move, i.e. $Z_{t+1} = Z_t$. It can be shown that, after an initial transient ("burn-in"), the accepted states are distributed according to π. In this work MCMC is used as a technique to solve optimization (7). Let us interpret the candidate values of θ as the outcomes of a random variable Θ. Consider the stochastic model formed by Θ and the M random vectors Y_θ^i. These vectors are conditionally independent given $\Theta = \theta$ and have identical distribution $p_\theta(y)$. Following (Lecchini-Visintini *et al.*, 2006), the Metropolis–Hastings algorithm may be implemented so to obtain joint realizations of $Z = (\Theta, Y_\Theta^1, \ldots, Y_\Theta^M)$ from the following distribution:

$$\pi(\theta, y_\theta^1, \ldots, y_\theta^M) \propto \prod_{i=1}^{M} J(y_\theta^i) p_\theta(y_\theta^i) \qquad (10)$$

where $J(y)$ is a reward function for outcome y:

$$J(y) \triangleq \mathbb{E}\left[\frac{Y_{\theta^*}}{\sqrt{{Y_{\theta^*}}^T Y_{\theta^*}}}\right]^T \frac{y}{\sqrt{y^T y}}. \qquad (11)$$

In this case the marginal distribution of Θ is:

$$\pi(\theta) \propto \left[\int J(y) p_\theta(y) dy\right]^M = J(\theta|Y_{\theta^*})^M.$$

Therefore, by choosing a sufficiently large M, this function can be peaked around the maximum points of $J(\theta|Y_{\theta^*})$. This allows to find approximate solutions to optimization (7)–(8) in that, after the burn-in period, the MCMC extractions of Θ will concentrate around the maxima of $J(\theta|Y_{\theta^*})$. In practice, the expectation in (11) is evaluated empirically offline by a formula analogous to (9). The procedure is formalized in Algorithm 1. A detailed discussion of the implementation may be found in (Lecchini-Visintini *et al.*, 2006) and references therein. A critical issue

Algorithm 1 *MCMC* algorithm

Set $t = 0$
Generate $\theta_0 \sim q(\theta)$
Generate $y_{\theta_0}^i \sim p_{\theta_0}(y)$, $i = 1, \ldots, M$
Set $\pi(\theta_0) = \prod_{i=1}^{M} J(y_{\theta_0}^i)$
repeat
  Set $t = t + 1$
  Generate $\tilde{\theta} \sim q(\theta)$
  Generate $y_{\tilde{\theta}}^i \sim p_{\tilde{\theta}}(y)$, $i = 1, \ldots, M$
  Set $\pi(\tilde{\theta}) = \prod_{i=1}^{M} J(y_{\tilde{\theta}}^i)$
  Set $\alpha(\theta_{t-1}, \tilde{\theta}) = \min\left\{\frac{\pi(\tilde{\theta})q(\theta_{t-1})}{\pi(\theta_{t-1})q(\tilde{\theta})}, 1\right\}$
  Sample a uniform $(0, 1)$ random variable U
  Set $\theta_t = \begin{cases} \tilde{\theta}, & \text{if } U \leq \alpha(\theta_{t-1}, \tilde{\theta}); \\ \theta_{t-1}, & \text{otherwise.} \end{cases}$
until False

is the choice of the proposal distribution q. The closer q is to π, the more effective is the random sampling of new candidates and the faster is the convergence of the chain. In the lack of reasonable approximations of π, a common choice is the random walk Metropolis Chain given by $q(\cdot|x_i) = N(x_i, \sigma^2)(\cdot)$. This choice guarantees that the algorithm keeps exploring the parameter space. However, the efficiency of the algorithm is very limited and lack of or slow convergence may occur. This increases the number of initial parameter samples that do not correspond to the target distribution and must be discarded, with significant waste of simulation time.

4. COMBINED GA–MCMC IDENTIFICATION

In (Lecchini-Visintini *et al.*, 2006) an iterative procedure has been proposed to adapt MCMC optimization to the distribution of interest and thus ensure faster chain convergence. According to this method, parameters are initially estimated by a random walk search. The search policy is then refined iteratively: several MCMC runs are executed, each with proposal distribution equal to a mixture of Gaussian distributions with means selected among the parameter estimates of the previous run. The results obtained by this implementation are presented in Figure 2. The accepted states (black crosses) are shown in scatter plots reporting the value of the first component of θ versus each of the other components. The white circles correspond to the real parameters.

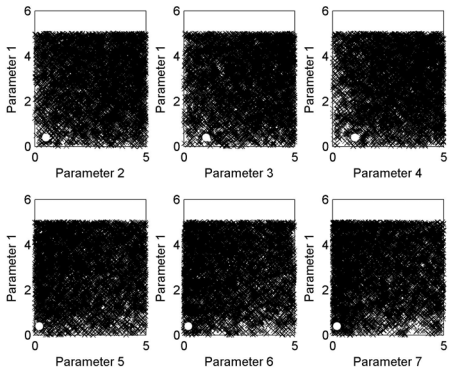

Fig. 2. Accepted states with MCMC.

Due to the large number of unknown parameters and the complexity of the model (i.e. of the target distribution), the chain suffered from lack of convergence. "Bad" and "good" parameters were equally accepted, yielding estimates that cover the entire parameter space. In principle, the obstacle may be overcome by simply letting the algorithm run long enough. However, because of the significant amount of time needed to simulate the model (about 5 seconds per run) a large amount of MCMC iterations is infeasible (20000 runs required 22 hours of computation). To improve convergence, the idea is to replace the initial random walk search of the parameter space with a convenient number of GA runs. While repeated runs of GA optimization may still be unable to guarantee consistent parameter estimates, they provide initial parameter guesses that can be used to "bias" the MCMC search towards the most relevant regions of the parameter space. In our implementation, the initial proposal distribution of MCMC is set to a mixture of Gaussian distributions with means given by the parameter estimates provided by H GA runs. Compared to random walk, this search distribution is already adapted to the actual shape of the target function. This induces the chain to converge in fewer iterations with significant savings in computational time. The coupled GA-MCMC procedure is summarized in Algorithm 2.

4.1 Results

The results by the implementation of the above algorithm with $N = 100$, $H = 200$, $M = 5$ and $\sigma = 0.5$ are presented in Figure 3. Comparison with Figure 2 allows one to appreciate the improvements. Accepted parameter values form clouds about the true values. In a realistic setting where the true parameters are unknown, this allows one to choose the parameters so that the behavior of the estimated model reflects the experimental data. Figure 4 illustrates the relevance of the accepted values

Algorithm 2 Coupled GA–MCMC algorithm

Run Genetic Algorithm optimization H times and store the results in a matrix G.
Set $t = 0$
Sample uniformly $h \in (0, H)$.
Generate $\theta_0 \sim q(\theta) = Gaussian(G(h), \sigma^2 I)$
Generate $y^i_{\theta_0} \sim p_{\theta_0}(y)$, $i = 1, \ldots, M$
Set $\pi(\theta_0) = \prod_{i=1}^{M} J(y^i_{\theta_0})$
repeat
  Set $t = t + 1$
  Sample uniformly $h \in (0, H)$.
  Generate $\tilde{\theta} \sim q(\theta) = Gaussian(G(h), \sigma^2 I)$
  Generate $y^i_{\tilde{\theta}} \sim p_{\tilde{\theta}}(y)$, $i = 1, \ldots, M$
  Set $\pi(\tilde{\theta}) = \prod_{i=1}^{M} J(y^i_{\tilde{\theta}})$
  Set $\alpha(\theta_{t-1}, \tilde{\theta}) = \min\left\{\frac{\pi(\tilde{\theta})q(\theta_{t-1})}{\pi(\theta_{t-1})q(\tilde{\theta})}, 1\right\}$
  Sample uniformly $U \in (0, 1)$
  Set $\theta_t = \begin{cases} \tilde{\theta}, & \text{if } U \leq \alpha(\theta_{t-1}, \tilde{\theta}); \\ \theta_{t-1}, & \text{otherwise.} \end{cases}$
until False

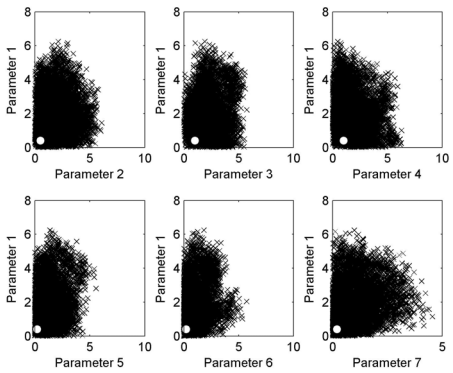

Fig. 3. Accepted states with coupled GA-MCMC.

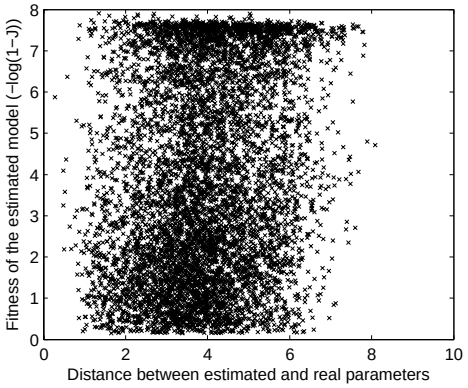

Fig. 4. Fitness of GA-MCMC accepted states.

in terms of their distance from the real parameter vector. The parameters closest to the real ones ($\theta^* = [0.4, 0.5, 1, 1, 0.2, 0.2, 0.2]$) are $\theta = [0.284, 0.492, 0.968, 1.053, 0.433, 0.125, 0.174]$ and one execution of the corresponding model is shown in Figure 5.

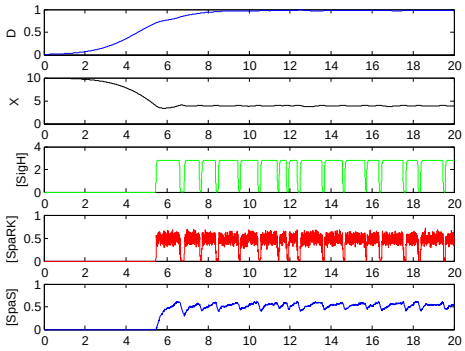

Fig. 5. One run of the estimated model with parameters closest to the real ones.

5. CONCLUDING REMARKS

We have presented a randomized technique for the parameter identification of a model of subtilin production by *Bacillus subtilis*. The method relies on the application of GA optimization as an initialization step for MCMC optimization. This allowed to speed up the execution of MCMC while preserving the theoretical convergence properties of MCMC that are not guaranteed by GA. Numerical simulations confirmed the goodness of our approach, in that MCMC optimization was largely improved by GA initialization. For the specific problem at hand, MCMC optimization provided evidence that estimation performance is affected by the weak identifiability of the parameters, i.e. a large domain of the parameter space explaining the data almost equally well. Developments of the work are twofold. On one hand, more extensive simulation and theoretical considerations may improve the definition of fit between experimental and synthetic data. On the other hand, application of the method to several other biological models is in our aims.

6. ACKNOWLEDGEMENTS

This research was supported by the European Commission under the project HYGEIA, NEST–4995.

REFERENCES

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

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

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

Davis, M.H.A. (1984). Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society, B* **46(3)**, 353–388.

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

Drulhe, S., G. Ferrari-Trecate, H. de Jong and A. Viari. (2006). Reconstruction of switching thresholds in piecewise-affne models of genetic regulatory networks. In: *Hybrid Systems: Computation and Control* (J. Hespanha and A. Tiwari, Eds.). number 3927 In: *LNCS*. Springer–Verlag. Berlin. pp. 184–199.

Ghosh, R. and C.J. Tomlin (2001). Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model. *Lecture Notes in Computer Science* **2034**, 232–246.

Hastings, W.K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**, 97–109.

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

Kouretas, P., K. Koutroumpas, J. Lygeros and Z. Lygerou (2006). Stochastic hybrid modeling of biochemical processes. In: *Stochastic Hybrid Systems* (C. G. Cassandras and J. Lygeros, Eds.). Vol. 24 of *Automation and Control Engineering Series*. CRC press.

Koutroumpas, K., E. Cinquemani, P. Kouretas and J. Lygeros (2006). Parameter identification for stochastic hybrid systems: An application to subtilin production by bacillus subtilis. Technical Report vol. AUT06-12. Automatic Control Laboratory (IfA, ETH).

Lecchini-Visintini, A., W. Glover, J. Lygeros and J.M. Maciejowski (2006). Monte carlo optimization for conflict resolution in air traffic control. *IEEE Transactions on Intelligent Transportation Systems* **7(4)**, 470–482.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller (1953). Equations of state calculations by fast computing machine. *J. Chem. Phys.* **21**, 1087–1091.