2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Optimal discovery of a stochastic genetic network

Robin L. Raffard, Ovidiu Lipan, Wing H. Wong and Claire J. Tomlin

Abstract— **In this paper, we present a parameter identification algorithm for the discovery of a genetic regulatory network. The genetic network is modeled, via a mechanistic approach, as a nonlinear stochastic regulatory network, in which transcription, translation and degradation processes are described as discrete stochastic events which depend nonlinearly on the number of molecules inside the cell. The system depends on several unknowns, namely the rates of transcription, the rates of translation and the degradation rates. Furthermore, the system is observable through the measure, at regular time intervals, of the factorial cumulants of the molecule counts. The unknown parameters are uncovered by studying the system output response to an arbitrary command input. The parameter search is posed as an optimization program, in which the cost function is the deviation between the observed factorial cumulants and the model output, and in which the constraint is the parameterized ordinary differential equation (ODE) governing the time evolution of the factorial cumulants. The optimization problem is solved via an adjoint-based quasi-Newton algorithm. The command input is found to have an important impact on the parameter search: Oscillatory input signals yield better parameter discovery than flat input signals. Finally, numerical results are presented for two systems: a Hill feedback and a Michaelis Menten process.**

I. Introduction

Cells can be viewed as robust stochastic molecular machines which respond to external cues via a sequence of biochemical reactions. Each reaction change the state of the cell. The transition from one state to another is probabilistic and can be quantitatively described by a set of transition probabilities. Using the words of S. Chandrasekhar [2], the molecular processes represent the "gradual unfolding of the transition probabilities". The "gradual unfolding" is governed by a Master equation [13], [1], [9]. The transition probabilities, which are the building blocks of the Master equation, can be identified by measuring the responses of the molecular system to different input signals. A theory based on the Master equation which explains the response of a genetic regulatory network to input signals was outlined in [1], [9]. For a single input-output pair, the theory was used to understand the response of mammalian cells to heat shocks [7]. For multiple inputs and multiple outputs, however, the theory of [1], [9] is not sufficient, and must be accompanied by a procedure that automatically estimates the transition probabilities using experimental data. The difficulty of estimating these transition probabilities stems (1) from the huge number of molecules involved in a genetic network and (2) from the stochasticity and nonlinearity of the processes. For example, for a system with only two molecules, besides the mean and the standard deviation of each molecule, the statistical correlation between the two molecules also needs to be modeled. Furthermore, the nonlinearity of the process will mix lower and higher order correlations in a nested system of equations. The number of variables thus increases dramatically with the number of components of the molecular system. In spite of these challenges, however, this paper presents a solution to the estimation problem of the transition probabilities of a Master equation. This is encouraging since it is known that the Master equation, although very simple to connect with the biochemical reactions, is difficult to solve and estimate.

In Section II, we will review the class of models which we use to describe the genetic network. In Section III, we will pose the problem of discovering the genetic network as an optimal control problem for ODEs. Section IV will demonstrate the methodology which we use in order to solve such optimization problems and Section V will outline the practical implementation of our method. In Section V, we will motivate the use of oscillatory input signals in order to enhance the discovery of the network. Finally, in Section VI, we will show numerical results for a Hill feedback and a Michaelis Menten process.

II. System Model: Nonlinear Stochastic Regulatory Network

In this section, we present the class of systems which we use to model the genetic network. The model relies on a mathematical framework developed by Lipan, Achimescu and Wong [9], [1], [8], and describes the system as a nonlinear stochastic network. If N denotes the number of genes involved in the network, the state variables of the system are the number of mRNA molecules and the number of protein products of the N genes. The dynamics of the system is driven by a continuous time Markov chain, in which the state transitions are governed by transcription, translation and degradation processes. The probability of occurrence of these transition events are given in a parametric form as nonlinear functions of the $2N$ state variables of the system. Finally,

This work was supported by the NIH Joint DMS/BIO/NIGMS Initiative to support research in the Area of Mathematical Biology, by the DARPA BioComputation grant, by a Stanford Bio-X award, and by a NSF-DMS grant.

R. Raffard was with the Department of Aeronautics and Astronautics at Stanford University, CA: rraffard@stanfordalumni.org. He is now with Barclays Global Investors, San Francisco, CA.

O. Lipan is with the Department of Physics at the University of Richmond, VA: olipan@richmond.edu.

W. Wong is with the Department of Statistics at Stanford University, CA: whwong@stanford.edu.

C. Tomlin is with the Department of Electrical Engineering and Computer Sciences at UC Berkeley, CA and with the Department of Aeronautics and Astronautics at Stanford University, CA: tomlin@eecs.berkeley.edu.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

the parameters of these transition probabilities need to be identified in order to reveal the genetic network.

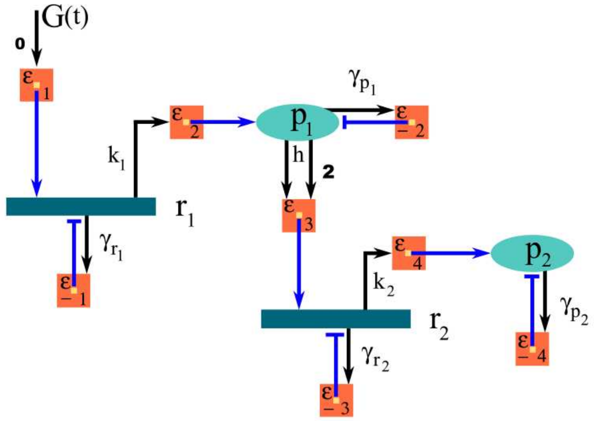

Fig. 1. Representation of the interaction between two genes as a network diagram. The protein product of gene 1 activates the transcription of gene 2. The command input $G(t)$ activates the transcription of gene 1.

As an example, consider the interaction between two genes, depicted in Figure 1. The variables r_1 and p_1 respectively represent the number of mRNA molecules and the number of protein products of gene 1. Similarly r_2 and p_2 represent the number of mRNA and the number of protein products of gene 2. Protein 1 is a transcription factor of gene 2, which means that protein 1 activates the transcription of gene 1 and therefore the production of mRNA 2. Gene 2 is assumed to have no effect on gene 1. The state variable of this system, denoted by $X = (r_1, p_1, r_2, p_2) \in \mathbb{N}_+^4$, evolves according to a continuous time Markov chain. The transitions of the Markov chain are denoted by $\varepsilon_i \in \{-1,0,1\}^4$, $i = -4,\dots,4$, and represent translation or transcription events. For instance, the transition from (r_1, p_1, r_2, p_2) to (r_1+1, p_1, r_2, p_2), denoted by $\varepsilon_1 = (1,0,0,0)$, represents the production of one molecule of mRNA 1 via the transcription of gene 1. The transition from (r_1, p_1, r_2, p_2) to (r_1, p_1, r_2, p_2+1), denoted by $\varepsilon_4 = (0,0,0,1)$, represents the production of one protein product of gene 2 via the translation of mRNA 2.

The transition probabilities T_{ε_i}, $i = -4,\dots,4$, are functions of the state variable X and the command input $G(t)$, and are expressed in a parametrized form. For example, the probability of transition T_{ε_2}, which is supposed to be proportional to the number of molecules r_1 is expressed as $T_{\varepsilon_2} = k_1 r_1$. Of course, the choice of the parametrization can be more general and can include some nonlinear terms. We denote by $\theta \in \mathbb{R}^d$ the vector containing all parameters. In the present example $\theta = (k_1, \gamma_{r_1}, \gamma_{p_1}, h, k_2, \gamma_{r_2}, \gamma_{p_2}) \in \mathbb{R}^7$.

Given the expression for these transition probabilities, the probability distribution of X can be derived as the solution of a so-called Master equation [9], [1]. If $P(x,t)$ denotes the probability that $X = x$ at time t, then at time $t+dt$

$$\begin{aligned} P(x,t+dt) \quad &= \quad P(x,t)\big(1 - dt\sum_{\varepsilon} T_\varepsilon(x,t)\big) \\ &\quad + \sum_{\varepsilon} P(x-\varepsilon,t)T_\varepsilon(x-\varepsilon,t)dt, \end{aligned} \tag{1}$$

in which the sums are taken over all possible transitions ε. Therefore the equation governing the probability distribution of X reads

$$\frac{\partial P(x,t)}{\partial t} = \sum_{\varepsilon} P(x-\varepsilon,t)T_\varepsilon(x-\varepsilon,t) - \sum_{\varepsilon} P(x,t)T_\varepsilon(x,t). \tag{2}$$

Note that $P(x,t)$ implicitly depends on the vector of parameters θ and the input $G(t)$ through the transition probabilities T_ε. Similarly to the resolution of partial differential equations, the complexity of solving this Master equation grows combinatorially with the dimension of the system. In practice, because of hardware memory constraints, it can be solved up to dimension of five. However from this equation we can derive an ordinary differential equation (ODE) governing the factorial cumulants of the molecule counts (the factorial cumulants of a random variable are analogous to its moments; however they present better convergence properties and are therefore used for this analysis). This ODE has the general form

$$H(\kappa(t),\theta)\frac{d\kappa(t)}{dt} = A(\kappa(t),\theta) + G(t)B(\kappa(t),\theta), \tag{3}$$

in which H is an $n \times n$ matrix whose entries are nonlinear function of $\kappa(t)$, and θ, A, and B are column vectors.

The detailed procedure leading to the derivation of the cumulants is explained in [1]. In particular, it relies on taking the $\mathscr{Z}$-transform of the state probability $P(X,t)$.

III. Parameter Identification Problem Formulation

With the technique of flow cytometry [12], it is possible to measure the status of tens of thousands of cells in a few seconds. Based on these samples, the entire statistics of the molecule numbers can be derived and, in particular, the factorial cumulants can be computed. Given a set of observations $\kappa_1^{\text{obs}}, \dots, \kappa_n^{\text{obs}}$, for the m first cumulants collected at times $t_1, \dots, t_n = T$, $n \in \mathbb{N}$, the problem of finding the unknown parameters of the system can be posed as the one of minimizing the mean square error between the observed cumulants and the outputs of the simulation

$$\boxed{\begin{array}{ll} \text{minimize} & J = \sum_{k=1}^{n} |\kappa_k^{\text{obs}} - \kappa(t_k)|^2 \\ \text{subject to} & H(\kappa(t),\theta)\frac{d\kappa(t)}{dt} = A(\kappa(t),\theta) \\ & \qquad + G(t)B(\kappa(t),\theta). \end{array} \qquad \text{(P.1)}}$$

Note that this approach of matching the first order factorial cumulants is similar to the so-called method of moments used in parameter identification of stochastic processes [5], [10]. Note also that in the present case, due to the large size of the sample population, the estimates of the factorial cumulants have converged and therefore the difference between the factorial cumulants and their estimates is only due to measurement noise. Because we assume white measurement noise, in (P.1) we minimize the least square error between the estimates of the factorial cumulants and the simulated factorial cumulants.

IV. Solution Method

The problem of finding the unknown parameters of the genetic network has been formulated through (P.1) as an optimization program involving ODEs. In this section, we show how such an optimization program can be (locally) solved efficiently using an adjoint-based quasi-Newton algorithm [4], [3] in the parameter space, $\mathbb{R}^d$. From a control theory standpoint, the algorithm consists of iteratively solving the Pontryagin necessary conditions for optimality. From an optimization theory point of view, it consists of a quasi-Newton method, in which the gradient of the objective function with respect to the vector of parameters θ is computed via the adjoint method in order to efficiently deal with the ODE constraint.

A. Gradient Computation

In order to perform the quasi-Newton method, we first need to compute the gradient of the objective function with respect to the vector of parameters. The computation of the gradient relies on two steps. First, the derivative of the cost function is calculated by application of the calculus of variations for ODEs. This step leaves the derivative of the cost function expressed as a function of the derivative of the factorial cumulants with respect to the parameters θ. Since the linear ODE governing the derivative of the factorial cumulants cannot be solved in closed form, the second step consists of eliminating the terms which depend on the factorial cumulant derivative in the expression of the cost function derivative. This second step is performed using the adjoint method [6], [11].

1) Calculus of Variations:

Proposition 1. *Under differentiability and growth conditions on A, B and H, the derivative of the cost function $J \in \mathbb{R}$ with respect to the vector of parameters $\theta \in \mathbb{R}^d$ in the direction $\hat{\theta} \in \mathbb{R}^d$ is given by*

$$\lim_{h\to 0} \frac{J(\theta + h\hat{\theta}) - J(\theta)}{h} = 2\sum_{k=1}^{n} \left(\kappa(t_k) - \kappa_k^{obs}\right)^\top \hat{\kappa}(t_k), \quad (4)$$

in which $\hat{\kappa} \in C^1(0,T;\mathbb{R}^m)$ is the solution of

$$\begin{aligned} &\nabla_\kappa H(\kappa(t),\theta)\hat{\kappa}(t)\frac{d\kappa(t)}{dt} + \nabla_\theta H(\kappa(t),\theta)\hat{\theta}\frac{d\kappa(t)}{dt} + \\ &H(\kappa(t),\theta)\frac{d\hat{\kappa}(t)}{dt} = \nabla_\kappa A(\kappa(t),\theta)\hat{\kappa}(t) + \nabla_\theta A(\kappa(t),\theta)\hat{\theta} \\ &\qquad + G(t)\nabla_\kappa B(\kappa(t),\theta)\hat{\kappa}(t) + G(t)\nabla_\theta B(\kappa(t),\theta)\hat{\theta}. \end{aligned} \quad (5)$$

2) Adjoint Method:

Proposition 2. *Let an arbitrary process $p \in PC^1(0,T,\mathbb{R}^m)$ be the unique solution of*

$$\begin{aligned} \frac{dH(\kappa(t),\theta)^\top p(t)}{dt} &= K_1(\kappa(t),\dot{\kappa}(t),\theta)^\top p(t) \\ -\left(\nabla_\kappa A(\kappa(t),\theta) + G(t)\nabla_\kappa B(\kappa(t),\theta)\right)^\top p(t), \end{aligned} \quad (6)$$

on $]t_k, t_{k+1}]$, $k = 0,\ldots,n-1$; with boundary conditions,

$$\begin{aligned} H(\kappa(t_n),\theta)^\top p(t_n) &= \kappa(t_n) - \kappa_n^{obs} \\ H(\kappa(t_k^-),\theta)^\top p(t_k^-) &= H(\kappa(t_k^+),\theta)^\top p(t_k^+) + \kappa(t_k) - \kappa_k^{obs}, \\ &\qquad k = n-1,\ldots,1, \end{aligned} \quad (7)$$

in which $t_0 = 0$ and

$$K_1(\kappa(t),\dot{\kappa}(t),\theta) = \begin{bmatrix} \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\kappa H_{1i}(\kappa(t),\theta) \\ \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\kappa H_{2i}(\kappa(t),\theta) \\ \vdots \\ \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\kappa H_{mi}(\kappa(t),\theta) \end{bmatrix}. \quad (8)$$

Furthermore, let us pose

$$K_2(\kappa(t),\dot{\kappa}(t),\theta) = \begin{bmatrix} \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\theta H_{1i}(\kappa(t),\theta) \\ \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\theta H_{2i}(\kappa(t),\theta) \\ \vdots \\ \sum_{i=1}^m \frac{d\kappa_i(t)}{dt}\nabla_\theta H_{mi}(\kappa(t),\theta), \end{bmatrix} \quad (9)$$

then

$$\begin{aligned} \lim_{h\to 0} \frac{J(\theta + h\hat{\theta}) - J(\theta)}{h} &= -\int_0^{t_n} p(t)^\top K_2(\kappa(t),\dot{\kappa}(t),\theta)\hat{\theta}\,dt \\ &+ \int_0^{t_n} p(t)^\top \left(\nabla_\theta A(\kappa(t),\theta) + G(t)\nabla_\theta B(\kappa(t),\theta)\right)\hat{\theta}\,dt. \end{aligned} \quad (10)$$

p is referred to as the adjoint process.

Corollary 1. *The gradient $\nabla J \in \mathbb{R}^d$ of the cost function $J \in \mathbb{R}$ with respect to the vector of parameters $\theta \in \mathbb{R}^d$ is*

$$\begin{aligned} \nabla J &= -\int_0^{t_n} K_2(\kappa(t),\dot{\kappa}(t),\theta)^\top p(t)\,dt \\ &\quad + \int_0^{t_n} \left(\nabla_\theta A(\kappa(t),\theta) + G(t)\nabla_\theta B(\kappa(t),\theta)\right)^\top p(t)\,dt. \end{aligned} \quad (11)$$

Because the computation of the gradient requires solving both the primal ODE (3) and the adjoint ODE (6), the complexity of forming the gradient is equal to the sum of the complexity of solving ODE (3) and ODE (6). Since the primal and the adjoint ODEs roughly have the same complexity [11], [6], the gradient computation complexity is twice the complexity of solving ODE (3). Note that the alternative method of finite difference yields a complexity of $2d$ times the complexity of solving ODE (3), which is d times the complexity of the adjoint method.

B. Quasi-Newton Method

With the gradient in hand, it is now possible to perform an effective descent algorithm, namely the quasi-Newton method, in which the Hessian, or second derivative of the objective function, is approximated via finite difference on the gradient [3], [4].

Algorithm 1 (Adjoint-based quasi-Newton algorithm).

```
Start by guessing an initial value for $\theta$ (for instance, take
  $\theta^{\text{guess}} = 0$) as well as an initial guess for the approximate
  Hessian $\widetilde{\nabla^2 J}$ (for instance take $\widetilde{\nabla^2 J}^{\text{guess}} = I_d$).
  Repeat
      1. Compute the quasi Newton step.
          a. Solve the governing ODE (3) for $\kappa$ based on
          the current value of $\theta$.
          b. Solve the adjoint ODE backward for $p$, based
          on (6) and on boundary conditions (7).
          c. Form the gradient $\nabla J$ according to (11).
          d. Form the Newton step $\Delta\theta_{\text{nt}} = -\widetilde{\nabla^2 J}^{-1}\nabla J$.
      2. Line search: compute the step size $\beta > 0$ such that
      $J(\theta + \beta\Delta\theta_{\text{nt}})$ is minimized.
      3. Update $\theta := \theta + \beta\Delta\theta_{\text{nt}}$.
      4. Update $\widetilde{\nabla^2 J}$ via finite gradient difference.
  until $|\nabla J^\top \widetilde{\nabla^2 J}^{-1}\nabla J|$ is smaller than stopping criterion.
  Return $\theta^{\text{opt}} = \theta$.
```

If Problem (P.1) is convex, then the quasi-Newton method converges to an optimal vector of parameters. However, if Problem (P.1) is not convex, the quasi-Newton algorithm only guarantees convergence to a local optimum [4], [3]. In general, the algorithm is tractable when the dimension of the governing ODE (3) is less than 1,000 and local convergence of the quasi-Newton method is typically obtained in 20 to 50 iterations.

V. Choice of the Input Signal: Oscillatory vs. Flat Input Signal

In order to enhance the discovery of the genetic pathway, we have the freedom to choose the network input signal $G(t)$, $0 \leq t \leq T$. Traditionally, flat command inputs have been used in biology under the form of growth factors. However, it has been argued that oscillatory signals could be implemented as well, and yield better system discovery [9]. In this section, we first motivate the use of oscillatory signals by an experimental analysis argument and then we show that oscillatory signals indeed yield better parameter identification.

A. Motivation

Frequently in wet lab experiments, a trend (or offset) is superimposed on the biological response. Such a trend can appear because of the cell growth, but it is not connected with the genetic network. In this case, problem (P.1) cannot be used to estimate the unknown parameters because the observed cumulants are erroneous. However, the trend can be eliminated by spectral analysis. Let $\kappa^{\text{trend}}(t)$ be an unknown constant: $\kappa^{\text{trend}}(t) = K^{\text{trend}}$ and let us suppose that the observed data $\kappa^{\text{obs}}(t)$ are given by the superposition of the genetic response $\kappa^{\text{true}}(t)$ and the trend $\kappa^{\text{trend}}(t)$

$$\kappa^{\text{obs}}(t) = \kappa^{\text{true}}(t) + \kappa^{\text{trend}}(t). \qquad (12)$$

In order to eliminate the trend, let us take the product of equation (12) with an arbitrary signal $a : t \to a(t) \in \mathbb{R}$

$$a(t)\kappa^{\text{obs}}(t) = a(t)\kappa^{\text{true}}(t) + a(t)\kappa^{\text{trend}}(t). \qquad (13)$$

Now if we observe the process at N different times such that: $t_1 = \frac{T}{N}, t_2 = \frac{2T}{N}, \ldots, t_N = T$, we can choose

$$a_n(t) = \cos(\frac{2\pi nt}{T}), \quad n = 1, \ldots, N-1, \qquad (14)$$

so that

$$\sum_{k=1}^{N} a_n(t_k)\kappa^{\text{trend}}(t_k) = 0, \quad \forall n = 1, \ldots, N-1. \qquad (15)$$

Instead of matching the observable time series data, the parameter identification problem can be posed as the one of matching the spectral components of the data. Namely,

$$\begin{array}{ll} \text{minimize} & J = \sum_{n=1}^{N-1} |\sum_{k=1}^{N} \cos(\frac{2\pi n t_k}{T})(\kappa^{\text{observed}}(t_k) - \kappa(t_k))|^2 \\ \text{subject to} & H(\kappa(t), \theta)\frac{d\kappa(t)}{dt} = A(\kappa(t), \theta) + G(t)B(\kappa(t), \theta). \end{array} \qquad \text{(P.2)}$$

In steady state, the spectral components of the factorial cumulants will not be zero if and only if an oscillatory signal input $G(t)$ is used to excite the system. Therefore, the use of oscillatory input signals is mandatory in order to eliminate trends from the experimental data.

B. Performance Measure

Even if no trend is present in the observed data, we will show in the next section that the use of oscillatory signals improves the performance of the genetic network discovery algorithm. For this purpose, we proceed as follows. We denote by $\theta^{\text{true}} \in \mathbb{R}^d$ the true values of the unknown parameters. We then deviate from this value by setting the initial guess for the parameters equal to their true value plus some noise (typically 500% noise): $\forall k \in \{1, \ldots, m\}, \theta_k^{\text{guess}} = 5^{\eta_k}\theta_k^{\text{true}}$, where $\{\eta_k | k = 1, \ldots, m\}$ are random variables uniformly distributed between -1 and 1. Finally we run the search algorithm. We compare the parameters returned, in the case of the oscillatory input signal and in the case of the flat command input, in terms of two metrics.

1) The first metric is the final cost J returned by the algorithm. It represents the quadratic deviation between the output of the simulation and the observed data.
2) The second metric $\delta = \sum_{k=1}^{d} \frac{|\theta_k - \theta_k^{\text{true}}|}{|\theta_k^{\text{true}}|}$ is the relative error between the returned parameters and the true parameters.

We expect that, regardless of the signal input, the cost function J will tend to zero when the number of iterations performed by the algorithm increases. However, the second metric, which represents the real performance of the search algorithm may be larger in one case than the other.

VI. Results

Results are given for the two following systems:

- A single gene auto-regulated by a Hill feedback
- A catalytic enzymatic process.

A. Hill Feedback

This system has two state variables: the number of mRNA molecules r and the number of proteins p of the gene under investigation. Transitions can occur if the gene is transcribed ($\varepsilon_1 = (1,0)$), if the gene is translated ($\varepsilon_2 = (0,1)$), if one mRNA molecule is degraded ($\varepsilon_{-1} = (-1,0)$) or if one protein molecule is degraded ($\varepsilon_{-2} = (0,-1)$). The probability of transitions are given by

$T_{\varepsilon_1} = G(t) + \dfrac{a_1 + a_2 p}{b_1 + b_2 p + b_3 p(p-1)}$	$T_{\varepsilon_{-1}} = \gamma_r r$
$T_{\varepsilon_2} = kr$	$T_{\varepsilon_{-2}} = \gamma_p p$

TABLE I. *Hill feedback:* Probability of transitions.

In the case of a flat input signal as well as in the case of the oscillatory input signal, the algorithm matches the desired data to a very good precision. This is indicated by the low value of the cost function J, always lower than or equal to $10e{-}6$. Furthermore, δ, the relative error between the true parameters and the parameters returned by the optimization algorithm, is quite informative. It is systematically smaller in the case of the oscillatory signal, which loosely means that the oscillatory signal reveals more information about the system than the flat signal. This difference is clear for the long time horizon case, in which the flat input only reveals the steady state of the system. For the short time horizon; independently of the input signal, the observations reveal part of the transient response, a great source of information about the system. Therefore, the difference between the oscillatory signal and the flat signal is not as large in this case.

	Step signal $G(t)=60$	Oscillatory signal $G(t)=50(1+\cos(t))$
$T = 4$ hours	$J = 1.0e{-}6$	$J = 2.7e{-}8$
$T = 96$ hours	$J = 5.1e{-}8$	$J = 1.5e{-}7$

TABLE II. *Hill feedback:* Optimal cost J returned by the search algorithm. The cost represents the quadratic deviation between the output of the simulation and the observed data.

The figures display the state variable over time. Figure 2 shows the initial guess for the state variable corresponding to $\theta = \theta^{\text{guess}}$. All other figures show the result of the algorithm (after the final iteration). The red crosses represent the observed data. The green line is the returned state variable and the blue circles (displayed in Figures 3 to 8) are the pseudo true data, which we can only observe at the observation times $t_k, k = 1,\ldots,n$, marked by the red crosses. Therefore, each red cross coincides by definition with a blue circle at times t_k.

	Step signal $G(t)=60$	Oscillatory signal $G(t)=50(1+\cos(t))$
$T = 4$ hours	$\delta = 3.2897$	$\delta = 2.1915$
$T = 96$ hours	$\delta = 2.5443$	$\delta = 0.0357$

TABLE III. *Hill feedback:* Relative error between the true parameters and the parameters returned by the search algorithm.

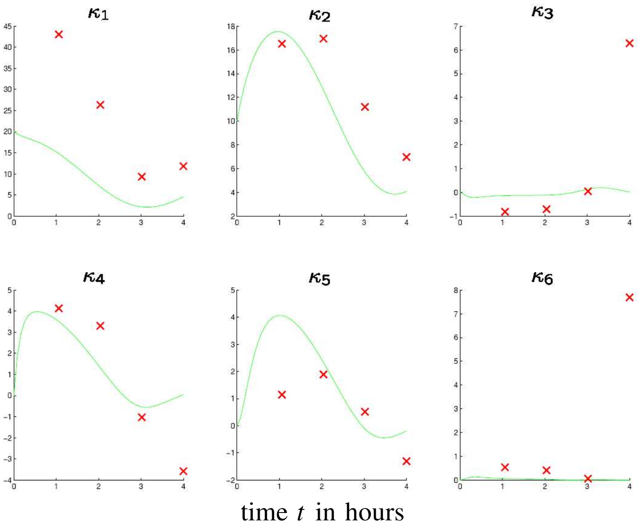

Fig. 2. *Hill feedback, oscillatory input, short time horizon:* Display of the initial guess of the six first factorial cumulants (green line), which have been used at the beginning of the search algorithm. The red crosses represent the observable data. The time horizon is four hours. $G(t) = 50(1+\cos(t))$.

B. Michaelis Menten Process

We now consider the enzymatic (E) catalytic process of transforming a substrate (S) into a product (P) through the formation of a complex (C):

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} \mathrm{E} + \mathrm{P}\,. \tag{16}$$

The state variables of the system are the number of enzymes E, the number of substrate molecules S, the number of complexes C and the number of products P. Nine transitions may occur: the production of one enzyme $\varepsilon_1 = (1,0,0,0)$ provoked by the command input, the natural degradation of one enzyme $\varepsilon_{-1} = (-1,0,0,0)$, the production of one molecule of substrata $\varepsilon_2 = (0,1,0,0)$, the degradation of one molecule of substrata $\varepsilon_{-2} = (0,-1,0,0)$, the four chemical reactions $\varepsilon_3 = (-1,-1,1,0)$, $\varepsilon_{-3} = (1,1,-1,0)$, $\varepsilon_4 = (1,0,-1,1)$, $\varepsilon_{-4} = (-1,0,1,-1)$, and finally the natural degradation of one molecule P, $\varepsilon_{-5} = (0,0,0,-1)$.

The results are shown in Table 5 and 6 and in Figures 7, 8, 9 and 10. They corroborate the results for the Hill

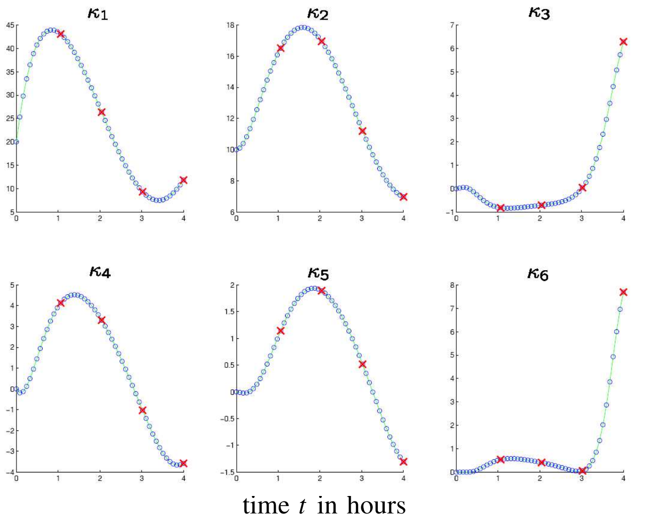

Fig. 3. *Hill feedback, oscillatory input, short time horizon:* Comparison between the output of the search algorithm (green line) and the output of the simulation corresponding to the true parameters (blue circles), for the six first factorial cumulants. The two outputs are very close to each other, which means that the search algorithm has properly converged. This will be the case for all results shown in Figures 4 to 8. The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is four hours. $G(t) = 50(1 + \cos(t))$.

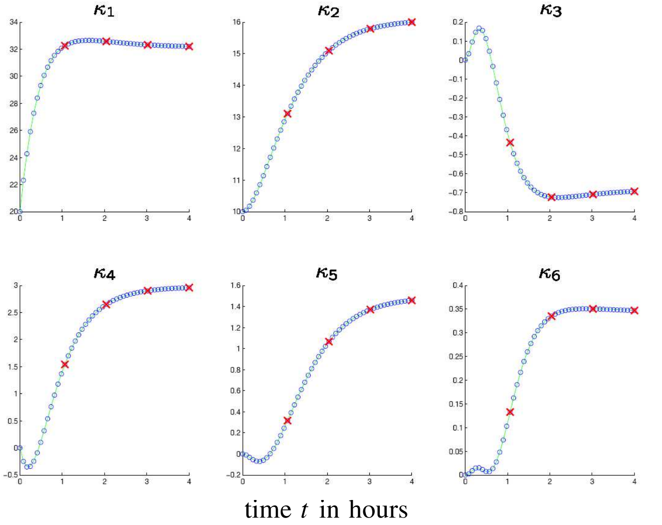

Fig. 4. *Hill feedback, flat input, short time horizon:* Comparison between the output of the search algorithm (green line) and the output of the simulation corresponding to the true parameters (blue circles). The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is four hours. $G(t) = 50$.

$T_{\varepsilon_1} = G(t)$	$T_{\varepsilon_{-1}} = \gamma_E E$	$T_{\varepsilon_2} = K_S$	$T_{\varepsilon_{-2}} = \gamma_S S$	$T_{\varepsilon_{-5}} = \gamma_P P$
$T_{\varepsilon_3} = k_1 ES$	$T_{\varepsilon_{-3}} = k_{-1} C$	$T_{\varepsilon_4} = k_2 C$	$T_{\varepsilon_{-4}} = k_{-2} EP$	

TABLE IV. *Michaelis Menten process:* Probability of transitions.

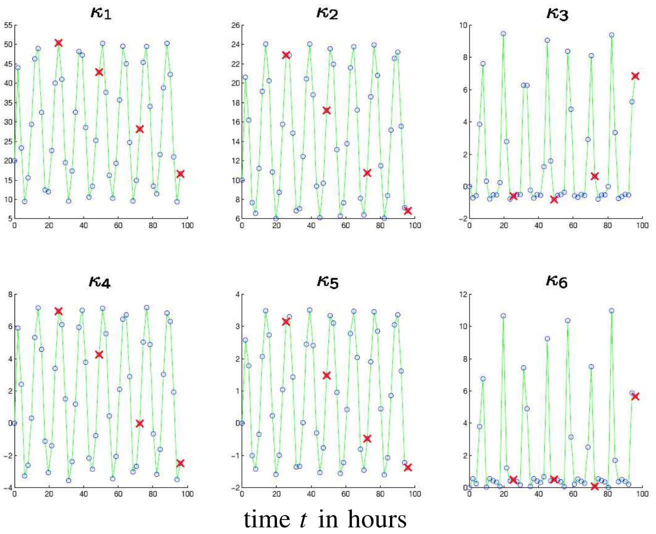

Fig. 5. *Hill feedback, oscillatory input, long time horizon:* Comparison between the output of the search algorithm (green line) and the output of the simulation corresponding to the true parameters (blue circles). The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is 96 hours. $G(t) = 50(1 + \cos(t))$.

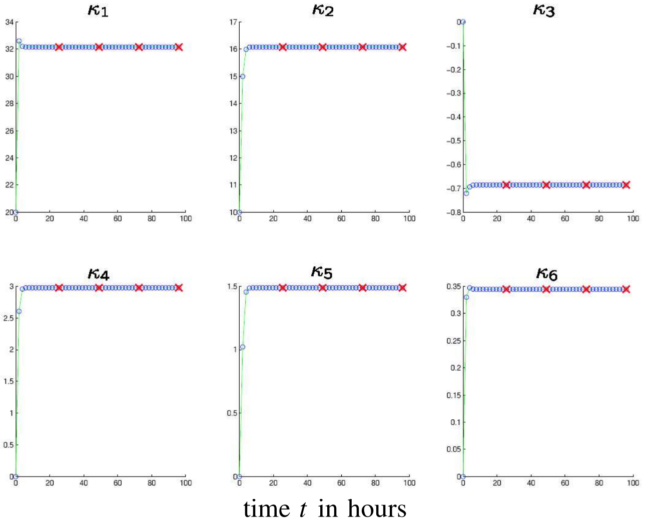

Fig. 6. *Hill feedback, flat input, long time horizon:* Comparison between the output of the search algorithm (green line) and the output of the simulation corresponding to the true parameters (blue circles). The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is 96 hours. $G(t) = 50$.

feedback example. In particular, oscillatory signals yield better parameter identification than flat input signals.

VII. Conclusions

With the emergence of new measurement tools, such as flow cytometry, allowing for simultaneous measurements of thousands of samples of gene expression levels, it is now possible to observe with high accuracy the statistics of genetic networks, and in particular it is possible to observe high order moments of gene molecule numbers. Using this precious information, optimal control theory allowed us to perform efficient parameter identification on a general class of stochastic genetic networks, introduced in [9], [1].

	Step signal $G(t)=200$	Oscillatory signal $G(t)=300(1+\cos(t))$
$T = 4$ hours	$J = 1.4e{-7}$	$J = 1.6e{-7}$
$T = 96$ hours	$J = 3.2e{-8}$	$J = 3.2e{-9}$

TABLE V. *Michaelis Menten process:* Optimal cost returned by the search algorithm. The cost represents the quadratic deviation between the output of the simulation and the observed data.

	Step signal $G(t)=200$	Oscillatory signal $G(t)=300(1+\cos(t))$
$T = 4$ hours	$\delta = 1.9e{-3}$	$\delta = 1.3e{-3}$
$T = 96$ hours	$\delta = 4.6e{-3}$	$\delta = 1.52e{-4}$

TABLE VI. *Michaelis Menten process:* Relative error between the true parameters and the parameters returned by the search algorithm.

Furthermore, the use of an oscillatory signal as an input in the genetic network was advocated in order to increase the amount of information revealed by the system and therefore to improve the network discovery.

References

[1] S. Achimescu and O. Lipan. Signal propagation in nonlinear genetic networks. *IEE Proceedings of Systems Biology*, 153:120–134, May 2006.

[2] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of Modern Physics*, 15:1–89, 1943.

[3] P. E. Gill and M. W. Leonard. Reduced-Hessian quasi-Newton methods for unconstrained optimization. *SIAM Journal on Optimization*, 12(1):209–237, 2001.

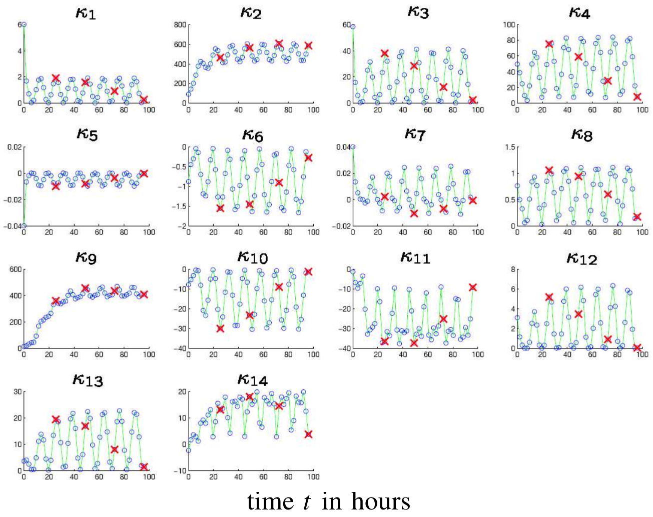

Fig. 7. *Michaelis Menten process with oscillatory input, long time horizon:* Comparison between the output of the search algorithm κ^{returned} (green line) and the output of the simulation corresponding to the true parameters (blue circles). The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is 96 hours. $G(t) = 300(1 + \cos(t))$.

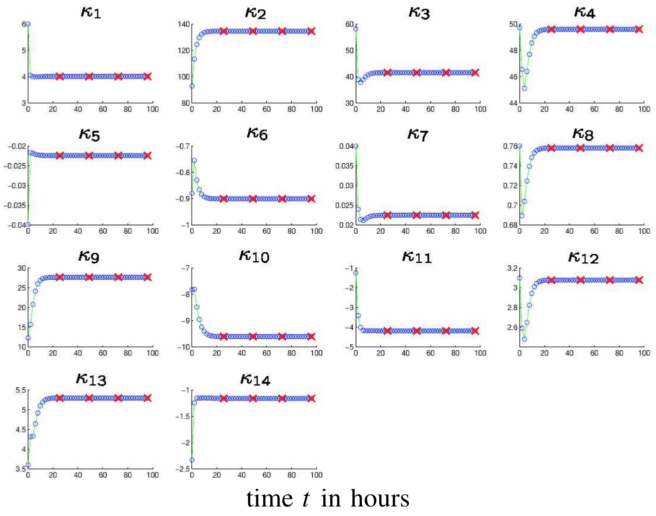

Fig. 8. *Michaelis Menten process with flat input, long time horizon:* Comparison between the output of the search algorithm (green line) and the output of the simulation corresponding to the true parameters (blue circles), for the 14 first factorial cumulants. The red crosses represent the observable data and are by definition on top of the blue circles at times t_k. The time horizon is 96 hours. $G(t) = 200$.

[4] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press. Harcourt Brace and Company, 1999.

[5] L. P. Hansen. Large sample properties of the generalized methods of moments. *Econometrica*, 50:1029–1054, 1982.

[6] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[7] O. Lipan, J.-M. Navenot, Z. Wang, Lei. Huang, and S. C. Peiper. Heat shock response in CHO mammalian cells is controlled by a nonlinear stochastic process. *PLoS Computational Biology*, e187.eor, 2007.

[8] O Lipan and W. H. Wing. Is the future biology shakespearean or newtonian? *Molecular BioSystems*, 2:411–416, 2006.

[9] O. Lipan and W. H. Wong. The use of oscillatory signals in the study of genetic networks. *Proceedings of the National Academy of Sciences of the USA*, 102(20):7063–7068, 2005.

[10] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, N.J., 2nd edition, 1999.

[11] R. L. Raffard. *Optimal control of systems governed by differential equations with applications in air traffic management and systems biology*. PhD thesis, Stanford University, December 2006.

[12] L. A. Sklar. *Flow cytometry for biotechnology*. Oxford University Press, 2005.

[13] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Amsterdam: North-Holland, 1992.