Backward stochastic differential equation approach to modeling of gene expression

Evelina Shamarova,[1,*] Roman Chertovskih,[2] Alexandre Ramos,[3] and Paulo Aguiar[4]

[1] *Departamento de Matemática, Universidade Federal da Paraíba, 58051-900, João Pessoa, Brazil*
[2] *Samara National Research University Moskovskoe shosse 34, Samara 443086 Russian Federation*
[3] *Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, Av. Arlindo Béttio 1000, 03828-00, São Paulo, SP, Brazil*
[4] *INEB - Instituto de Engenharia Biomédica i3S - Instituto de Investigação e Inovação em Saúde, Rua Alfredo Allen 208, 4200-135 Porto, Portugal*

(Dated: August 5, 2016)

In this article, we introduce a novel backward method to model stochastic gene expression and protein level dynamics. The protein amount is regarded as a diffusion process and is described by a backward stochastic differential equation (BSDE). Unlike many other SDE techniques proposed in the literature, the BSDE method is backward in time; that is, instead of initial conditions it requires the specification of endpoint ("final") conditions, in addition to the model parametrization. To validate our approach we employ Gillespie's stochastic simulation algorithm (SSA) to generate (forward) benchmark data, according to predefined gene network models. Numerical simulations show that the BSDE method is able to correctly infer the protein level distributions that preceded a known final condition, obtained originally from the forward SSA. This makes the BSDE method a powerful systems biology tool for time reversed simulations, allowing, for example, the assessment of the biological conditions (e.g. protein concentrations) that preceded an experimentally measured event of interest (e.g. mitosis, apoptosis, etc.).

PACS numbers: 02.50.Fz, 87.10.Mn, 07.05.Tp, 87.16.Yc

I. INTRODUCTION

Gene regulatory networks involving small numbers of molecules can be intrinsically noisy and subject to large protein concentration fluctuations [1, 2]. This fact substantially limits the ability to infer the causal relations within gene regulatory networks and the ability to understand the mechanisms involved in healthy and pathological conditions. A large interest has been raised in developing tools for gene regulatory network inference [3, 4] acknowledging the noisy/stochastic properties of experimental data [5–7], in parallel with studies addressing the prospective, forward, simulation of stochastic equations describing biochemical reactions [8]. There is however another context which, despite its relevance as a tool to better understand intracellular dynamics, has received little attention from a mathematical modeling perspective. That is the situation where the basic gene regulatory network is known, together with a present distribution of molecules/proteins, and one wants to infer the previous molecules distributions that gave rise to the observed data. This is the case, for example, of a sample of necrotic cells where the concentration distributions for the relevant molecules can be calculated, and one would like to infer the previous concentrations that gave rise to the necrotic condition. In this context, the problem can be addressed with backward stochastic differential equations.

In what follows, we present a method to model gene expression based on backward stochastic differential equations. We consider a gene regulatory network, where the stochastic variables are the amounts of proteins that are expressed from the genes of the network. To illustrate our method, we apply it to three simple gene networks: a positive self-regulating gene, which is the simplest network, and two networks composed by two and five interacting genes. To generate data to test and validate our approach we use Gillespie's stochastic simulation algorithm, referred to below as SSA, for simulation of biochemical reactions [8]. From the trajectories of multiple simulations, the SSA provides the distribution of protein amounts at a fixed final time, as well as at some fixed moments of time prior to the final. For realization of the SSA we used the COPASI software [9]. The network models used in the BSDE and the SSA simulations were taken the same. The BSDE method, which requires the final distribution as the input data, was applied to perform a simulation backwards in time. Importantly, at the end of the backward simulation we arrive at some deterministic value for the number of proteins which is very close to the SSA initial condition. Since in many applications the initial protein amounts are not known, and are, in fact, the goal of the study, we believe that our approach can be a useful tool in systems biology.

A. BSDE versus SDE

BSDEs were introduced by Bismut in 1973 [10], and over the last twenty years have been extensively studied by many mathematicians (e.g. [11], [12]). A BSDE

* evelina@mat.ufpb.br

considered in this work takes the form

$$\eta_t = h(B_T) - \int_t^T f(\eta_s)\, ds - \int_t^T z_s\, dB_s,\ t \in [t_0, T], \quad (1)$$

where B_t is a real-valued Brownian motion, and describes the dynamics of the vector of protein amounts η_t. One may think that (1) is equivalent to a usual (forward) SDE, since, similar to ODEs, knowing the final condition instead of the initial should lead to an equivalent problem. However, this is not the case if we require the solution to be adapted with respect to the forward Brownian filtration $\mathcal{F}_t = \sigma\{B_s, s \in [t_0, t]\}$, where σ stands for the σ-algebra generated by the random variables B_s up to time t. The requirement for the pair (η_t, z_t) to be $\mathcal{F}_t$-adapted implies that, under some assumptions, BSDE (1) has a unique solution pair (η_t, z_t) [11]. Therefore, (1) is a different object than the traditional (forward) SDE. One may not be convinced why we should require from the solution η_t to be $\mathcal{F}_t$-adapted. Gillespie [13] proposed to model the dynamics of amounts of molecules changing during a chemical reaction by a forward SDE known as the Chemical Langevin Equation

$$\eta_t = \zeta + \int_{t_0}^t f(s, \eta_s)\, ds + \int_{t_0}^t z_s\, dB_s,$$

where ζ is the initial condition at time t_0. However, if η_t solves this equation, then it is automatically $\mathcal{F}_t$-adapted. The process z_t also must be $\mathcal{F}_t$-adapted to ensure the existence of the stochastic integral. Therefore, the requirement for the solution pair (η_t, z_t) to BSDE (1) to be $\mathcal{F}_t$-adapted is a natural consequence of the Langevin dynamics. *In this article, we propose to use a BSDE for modeling simple gene expression networks due to its property to have a pair of stochastic processes (η_t, z_t) as the unique solution.* The latter fact is important since the noise generating process z_t is usually unknown.

B. Determining the final condition

The final condition for (1) is required to have the form $h(B_T)$, where T is the fixed final time. We construct numerically a piecewise linear function h so that $h(B_T)$ approximates a given final distribution provided by the SSA simulation. In practice, to obtain a distribution of protein amounts at time T, a large population of genetically identical cells is usually considered.

C. Diffusion process approximation

We note that the stochastic process describing the protein number is an integer-valued pure-jump process which may change its values by ± 1 at time, while the solution to (1) is a continuous process. However, assuming that the number of proteins of each type is sufficiently larger than 1, and the waiting times until the next synthesis or degradation are much smaller than the length of the interval $[t_0, T]$, we can model the synthesis and degradation of proteins employing continuous diffusion processes, i.e. by BSDEs with Brownian drivers as (1). A diffusion process approximation for the dynamics of amounts of molecules was undertaken, for example, in [13–15].

II. THE BSDE METHOD

In what follows, we describe the BSDE method to model gene expression networks. Specifically, we model the dynamics of protein amounts expressed by the genes of a gene regulatory network. In our simulation, the protein synthesis and degradation occurs on the time interval $[t_0, T]$. The input data for the BSDE method is the protein number distribution at time T. The amount of proteins is modeled by a continuous $\mathbb{R}^n$-valued diffusion process $\eta_t = (\eta_1(t), \eta_2(t), \ldots, \eta_n(t))$, where n is the amount of species, or types of proteins expressed by the genes of the network, and $\eta_i(t)$ is the amount of the i-th type of protein at time t.

It is assumed that the distribution of the final amounts of proteins η_T is given in the form of a histogram H. We construct an $\mathbb{R}^n$-valued function h so that the distribution of $h(B_T)$ produces a histogram approximately identical to H. The random variable $h(B_T)$ is, therefore, taken as the final data approximation. The piecewise linear continuous increasing function h has the form

$$h(x) = \sum_{i=1}^{N} \mathbb{I}_{(r_i, r_{i+1}]}(k_i x + b_i), \quad (2)$$

where $\mathbb{I}$ is the indicator function. Let l_i, $i = 1, 2, \ldots$, be the bin ends of the given histogram H, and p_i be the bin probabilities. We produce 20000 realizations of the random variable B_T. The endpoint r_1 is the smallest of the realizations of B_T. Suppose we constructed the endpoint r_i. Then, we choose the next endpoint r_{i+1} so that the probability that a Gaussian random variable with mean zero and variance $\sqrt{T}$ belongs to $(r_i, r_{i+1}]$ equals to p_i. We compute k_i as $(l_{i+1} - l_i)/(r_{i+1} - r_i)$, while b_i is chosen by the continuity argument.

In our model, the evolution of η_t is governed by the following BSDE

$$\eta_t = h(B_T) - \int_t^T f(\eta_s)\, ds - \int_t^T z_s\, dB_s,\ t \in [t_0, T], \quad (3)$$

where f is an $\mathbb{R}^n$-valued function representing the synthesis/degradation rate of the proteins. Here the transcription and translation are treated effectively as a single process. In our approach the operons are neglected or have their corresponding concentrations of products labeled by one of the proteins that they codify. In other words, we assume that the different mRNA's transcribed from the gene are translated at the same rate. The synthesis/degradation rate f (propensity function) is discussed in detail in Section III A.

BSDE (3) is solved numerically by the following algorithm. The theory of BSDEs [12] implies that if $\theta(t,x)$ is a solution to the final value problem

$$\begin{cases} \partial_t\theta(t,x) + \frac{1}{2}\theta_{xx}(t,x) - f(\theta(t,x)) = 0, \\ \theta(T,x) = h(x), \quad x \in \mathbb{R}, \end{cases} \tag{4}$$

then $\eta_t = \theta(t, B_t)$ and $z_t = \nabla\theta(t, B_t)$ is the unique solution to (3) adapted with respect to the Brownian filtration $\mathcal{F}_t = \sigma\{B_s, 0 \leqslant s \leqslant t\}$. Note that the function h may not be unique since there are many functions mapping $[r_i, r_{i+1}]$ to $[l_i, l_{i+1}]$. However, by continuity, all these functions are close to each other in the supremum norm. The solutions to (4) corresponding to close functions h will be also close to each other in the same norm.

Once a numerical solution to (4) is found, we simulate a sufficient number of Brownian motion trajectories B_t starting at zero at time t_0 and obtain the trajectories η_t as $\theta(t, B_t)$. In our simulation, we took 20000 trajectories of B_t. The noise can be computed as the stochastic integral $\int_t^T \nabla\theta(s, B_s)dB_s$, however, as mentioned earlier, in this work we are only interested in the protein amount process η_t.

By doing the time change $\tilde{\theta}(t,x) = \theta(T-t,x)$ we transform (4) to a Cauchy problem with the initial condition $\tilde{\theta}(0,x) = h(x)$. Note that, by (2), the function h is defined only on a compact interval $[r_1, r_{N+1}]$ which is the support for all the realizations of B_T. The values of h outside of this interval do not affect the solution to (3). Therefore, we can extend h to the whole real line $\mathbb{R}$ so that the extended function is continuous and its derivative vanishes outside of a compact interval $[a, b]$. Therefore, in practice, instead of (4) we solve the following initial-boundary value problem:

$$\begin{cases} \partial_t\tilde{\theta}(t,x) - \frac{1}{2}\tilde{\theta}_{xx}(t,x) + f(\tilde{\theta}(t,x)) = 0, \\ \tilde{\theta}(0,x) = h(x), \\ \tilde{\theta}_x(t,a) = \tilde{\theta}_x(t,b) = 0. \end{cases} \tag{5}$$

Problem (5) is solved numerically using the finite-difference discretization with the implicit treatment of the linear terms (the Crank-Nicolson method) and the explicit treatment of the nonlinear terms. In all computations the time step is taken 10^{-4}, and the uniform spatial grid (including the boundaries) is constituted of 1025 points. We verified that doubling the spatial and the temporal resolutions shows no qualitative difference.

III. NUMERICAL REALIZATION AND RESULTS

We employed the SSA to produce data for comparison with the BSDE method. Specifically, we performed a number of numerical simulations using the software COPASI [9], which implements the SSA. The following three cases were simulated: a self-regulating gene, and networks of two and five interacting genes. For all the networks, the distributions of protein numbers produced by the two methods, were compared at two middle time points by analyzing visually the corresponding histograms plotted jointly, and by comparing the means and the standard deviations. Also, we studied how precise the initial protein numbers for the SSA were recovered by the BSDE method.

In all simulations, the time is measured in seconds. We used the default options for numerics of the SSA implemented in COPASI.

At time $T = 200$, the distribution obtained by the SSA for each type of protein is used to produce a histogram which we take as the input data for our method.

A. Propensity functions

In (3), the function $f(\eta_t)$ represents (phenomenologically) the protein synthesis and degradation. The degradation rate for the i-th protein is a linear function with the rate constant ρ_i, while ν_i stands for the maximum synthesis rate. The regulatory effects of a given set of proteins onto the i-th gene are represented by a sigmoidal function multiplied by ν_i. This sigmoidal function is derived from the diffusion limited Arrhenius rate law (see [16]). Namely, the synthesis/degradation rate (propensity function) takes the form

$$f_i(\eta) = \nu_i \frac{\exp(\Theta_i)}{1+\exp(\Theta_i)} - \rho_i\eta_i, \tag{6}$$

where the first term is the rate of proteins synthesis, and the second term is the proteins degradation rate. Here Θ_i represents the net regulatory effect of the proteins regulating the activity of the i-th gene and is defined as follows: $\Theta = A\eta$, where $\Theta = (\Theta_1, \ldots, \Theta_n)$ and $A = \{A_{ij}\}$ is the $n \times n$ matrix whose element A_{ij} describes the regulation of the i-th gene by the j-th protein. Specifically, A_{ij} can be negative, positive, or null, indicating repression, activation or non-regulation of the i-th gene by the j-th protein. If Θ_i goes to the negative infinity, the synthesis rate tends to zero, and it tends to its maximum value ν_i for Θ_i going to the positive infinity.

In case of one protein ($n = 1$), we consider a positive self-regulating gene whose synthesis rate is given by a Hill function multiplied by the maximum protein synthesis rate ν [17], and the degradation rate is a linear function with the rate constant ρ:

$$f(\eta) = \nu\frac{a\eta^2}{1+a\eta^2} - \rho\eta. \tag{7}$$

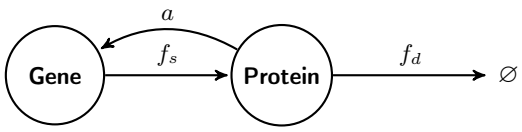
Here a is a positive constant indicating the strength of the self-regulation.

B. Self-regulating gene

We started by analyzing the simulation results for a self-regulating gene. The synthesis/degradation rate

$f(\eta)$, given by (7), was taken with the parameters $a = 1$, $\nu = 1$, and $\rho = 0.001$.

The network model for the self-regulating gene is shown on the diagram below with f_s and f_d standing for the synthesis and degradation rates, respectively.



The SSA simulation with 20000 trajectories started at time $t_0 = 0$, and the values of protein numbers for each trajectory were registered at times $t = 50, 100$, and 200. Next, we represented the SSA data at time $T = 200$ in the form of a histogram H. Using our technique of final data approximation described in Section II, we found a function h, so that 20000 realizations of the random variable $h(B_T)$ give rise to a histogram very close to H. In Figure 1, we show the histogram H for the SSA data and its approximation $h(B_T)$ at time $T = 200$ which demonstrates that our final data approximation technique is quite precise. We took $h(B_T)$ as the final data for BSDE

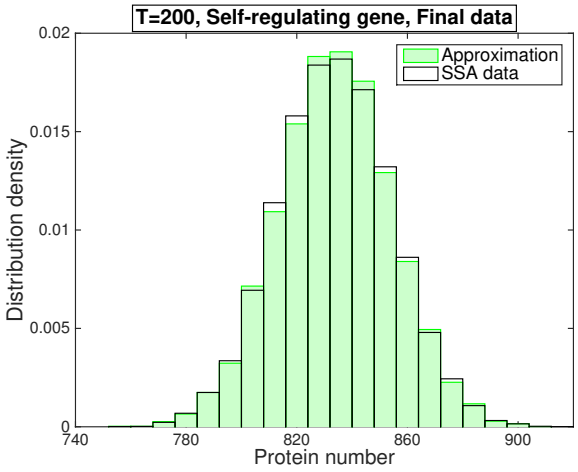


FIG. 1. Histograms for the SSA data and for the approximation $h(B_T)$ at time $T = 200$.

(3) and applied the BSDE method to simulate 20000 trajectories backwards in time starting from $T = 200$. The distributions of the protein numbers were determined at $t = 50$ and $t = 100$, and the corresponding histograms were plotted jointly with histograms for the SSA data as shown in Figure 2.

The means μ and the standard deviations σ for the data obtained by the both methods are presented in Table I. Although obtained by very different methods, the means and the standard deviations are in good agreement. The percent difference errors were computed as follows:

$$\text{Err}\,\mu = |(\mu_{\text{SSA}} - \mu_{\text{BSDE}})/\mu_{\text{SSA}}|,$$
$$\text{Err}\,\sigma = |(\sigma_{\text{SSA}} - \sigma_{\text{BSDE}})/\sigma_{\text{SSA}}|.$$

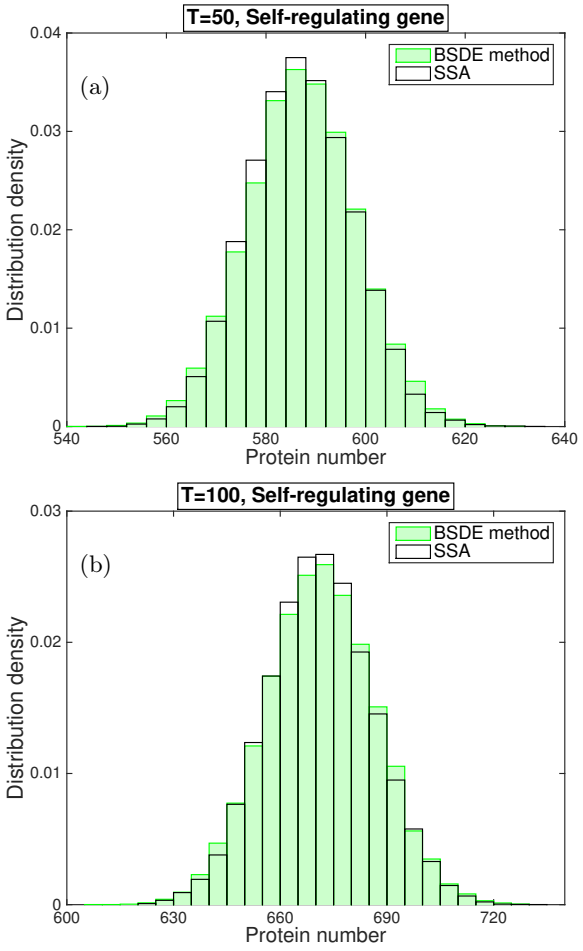


FIG. 2. Distributions of protein numbers for the self-regulating gene at $t = 50$ (a) and $t = 100$ (b) for the BSDE method and the SSA.

TABLE I. The means μ and standard deviations σ for the distribution of protein numbers for a self-regulating gene computed at $t = 0$, 50, and 100. The data obtained by the BSDE method are in the second and the third columns, and the data obtained by the SSA are in the fourth and the fifth columns. The last two columns present the percent difference errors.

	BSDE		SSA		%Errors	
Time	μ	σ	μ	σ	Err μ	Err σ
0	500.75	0	500	0	0.15%	–
50	587.13	10.99	586.44	10.53	0.11%	4.35%
100	671.37	15.30	670.78	14.78	0.08%	3.51%
200	833.80	20.46	833.27	20.52	0.06%	0.31%

Some trajectories of the BSDE solution η_t, representing the evolution of the number of proteins generated by a self-regulating gene, are shown in Figure 3. This is an illustrative example of what the output of the BSDE method looks like, and how the trajectories of η_t return to the same point which is close to 500. This is a good approximation of the protein number that we used as the starting point for the SSA, and therefore, the prediction

of this number by the BSDE method is very precise.

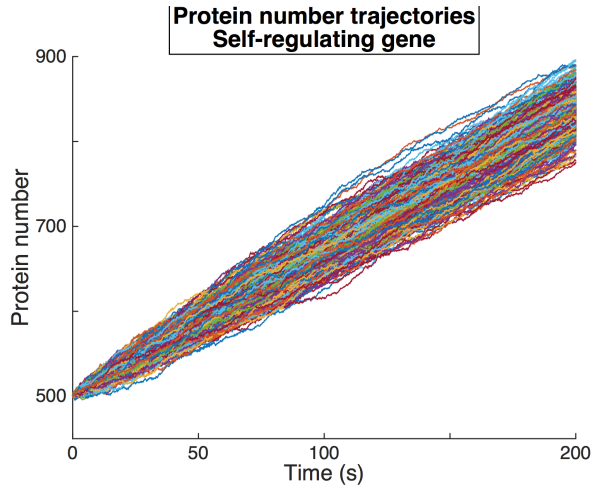
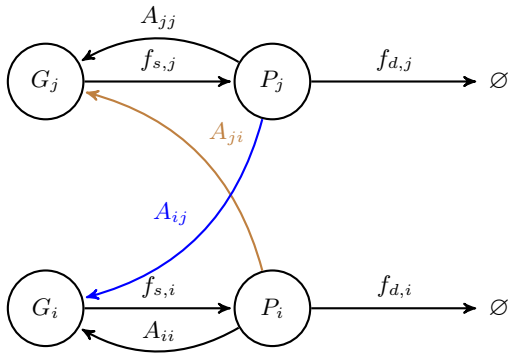


FIG. 3. Trajectories of the stochastic process η_t, describing the protein number for the self-regulating gene, obtained by the BSDE method.

C. Networks of interacting genes

We tested our method for gene regulatory networks consisting of two and five genes. The network models were taken as in the diagram below.



Here gene i, denoted by G_i, generates proteins of type i, which we denote by P_i, with the synthesis rate $f_{s,i}$ given by the first term in (6). Proteins P_i disappear with the degradation rate $f_{d,i}$ given by the second term in (6). Proteins P_i have a regulatory effect on gene j (denoted by G_j), which is represented by the regulation coefficient A_{ji}. This holds for any pair $G_i - P_j$. In particular, it is assumed that gene G_i generates only protein P_i, i.e. gene G_i cannot generate proteins of other types. This means that the number of genes equals to the number of protein types, i.e. to the dimension of the random vector $(\eta_1, \ldots, \eta_n)$, where n is either two or five.

For the network of two genes we considered the following values of parameters: $\nu_1 = 0.5$, $\nu_2 = 1$, $\rho_1 = 10^{-3}$, $\rho_2 = 5 \cdot 10^{-4}$, $A_{11} = 2$, $A_{12} = -1$, $A_{21} = 1$, $A_{22} = 0$. For the network of five genes we considered $\nu = (0.5, 1, 1, 1, 0.5)$, $\rho = (10^{-3}, 5 \cdot 10^{-4}, 10^{-3}, 5 \cdot 10^{-4}, 10^{-3})$, $A_1 = (2, -1, 0, 1, 0)$, $A_2 = (1, 0, 0, 0, 2)$, $A_3 = (1, 0, 1, 0, 0)$, $A_4 = (0, 0, 1, 1, 1)$, $A_5 = (0, 1, 0, 0, 1)$, where the i-th component of ν is ν_i, the i-th component of ρ is ρ_i, and A_i denotes the i-th line of the matrix A, $i = 1, 2, 3, 4, 5$. The final time T equals to 200 in both simulations.

The numerical algorithm was exactly the same as for the self-regulating gene. The number of trajectories in both methods was taken 20000. Specifically, the SSA simulation started at $t_0 = 0$, and the values of protein numbers for each trajectory were determined at times $t = 50, 100$, and 200. The distribution at final time $T = 200$ was approximated by $h(B_T)$, and the BSDE method provided the distributions at $t = 50$ and 100, which were compared with the distributions of the SSA data. In Figures 4 and 5 we show the distributions at $t = 50$ and 100 for some genes of the networks of two and five genes, respectively.

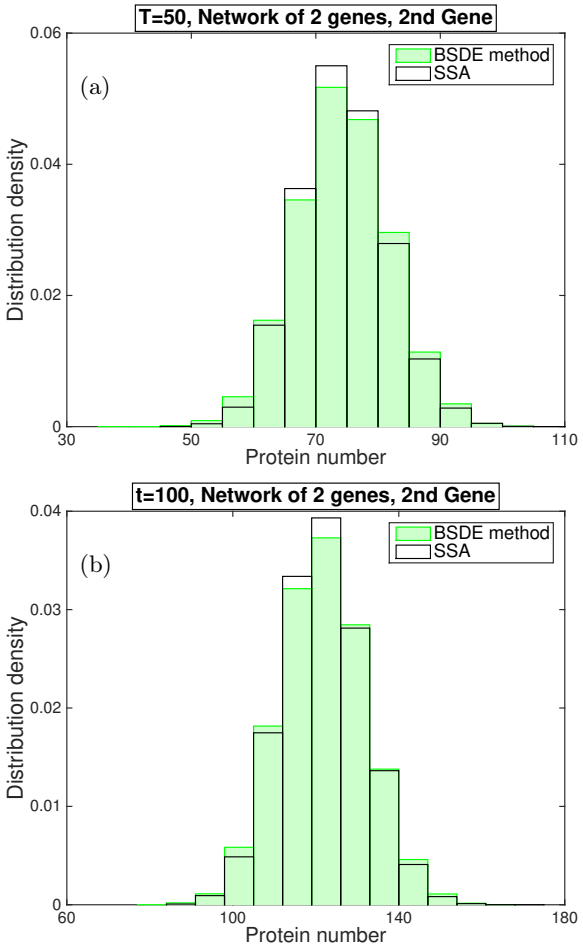


FIG. 4. Distributions of protein numbers at $t = 50$ (a) and $t = 100$ (b) for the 2nd gene of the network of two genes. The distributions are obtained by the BSDE method and the SSA.

We compared the means and the standard deviations

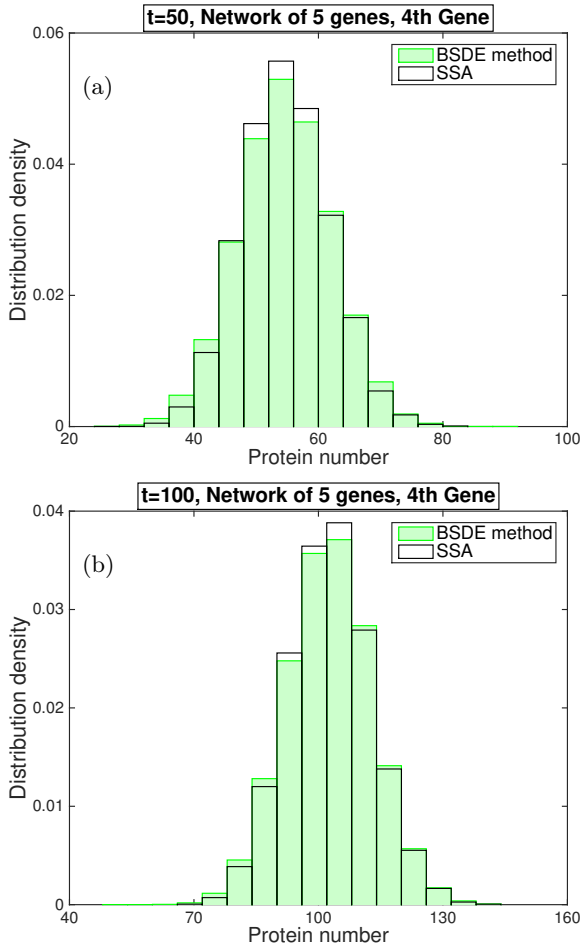


FIG. 5. Distributions of protein numbers at $t = 50$ (a) and $t = 100$ (b) for the 4th gene of the network of five genes. The distributions are obtained by the BSDE method and the SSA.

at $t = 50$ and 100 for the data obtained by the both methods. At final time $T = 200$ we compared the means and the standard deviations obtained by the SSA and by our technique of final data approximation. The results are represented in Tables II and III.

D. Prediction of the initial value

As we mentioned before, the BSDE method can be used to approximate the initial number of proteins. Since the solution to (3) can be represented as $\eta_t = \theta(t, B_t)$, where θ is the solution to final value problem (4), then, as it is implied by the BSDE method, the initial protein number η_0 is deterministic and equals to $\theta(0, 0)$. Tables I, II, and III show that the BSDE method provides a good approximation for the initial number of proteins used as an initial condition in the SSA. The percent difference error is the biggest, $6,89\%$, when the initial number of proteins is 5 (see Table III), which is the smallest considered in our simulations. The percent difference error decreases when the initial protein number increases, and it equals to $0,15\%$ when we deal with large initial protein numbers as in the case of the self-regulating gene (see Table I).

TABLE II. The first four columns contain the means μ and the standard deviations σ for the protein numbers of the network of 2 genes at $t = 0$, 50, and 100 obtained by the BSDE method and the SSA. At $T = 200$ we present μ and σ obtained using the final data approximation technique in comparison with the SSA data. The last two columns conintain the percent difference errors.

Network of 2 genes

	BSDE		SSA		%Errors	
t	μ	σ	μ	σ	Err μ	Err σ
			1st gene			
0	50.22	0	50.00	0	0.44%	–
50	72.24	6.24	71.88	5.14	0.50%	21.58%
100	92.99	8.39	92.75	7.21	0.25%	16.41%
200	131.85	10.81	131.23	10.94	0.47%	1.16%
			2nd gene			
0	25.43	0	25.00	0	1.74%	–
50	74.29	7.54	73.79	7.10	0.67%	6.22%
100	121.69	10.39	121.28	9.90	0.33%	4.93%
200	213.48	13.87	212.84	13.94	0.30%	0.58%

TABLE III. The data representation is the same as in Table II.

Network of 5 genes

	BSDE		SSA		%Errors	
t	μ	σ	μ	σ	Err μ	Err σ
			1st gene			
0	50.80	0	50	0	1.61%	–
50	72.70	5.75	71.90	5.19	1.11%	10.78%
100	93.54	7.72	92.86	7.19	0.73%	7.37%
200	132.23	9.89	131.73	9.92	0.38%	0.32%
			2nd gene			
0	25.51	0	25	0	2.04%	–
50	74.25	7.52	73.78	7.14	0.63%	5.31%
100	121.79	10.35	121.39	9.93	0.33%	4.21%
200	213.42	13.94	212.92	13.98	0.23%	0.28%
			3rd gene			
0	10.58	0	10	0	5.83%	–
50	58.82	7.77	58.31	6.98	0.89%	11.27%
100	104.72	10.43	104.16	9.86	0.53%	5.73%
200	189.94	13.36	189.43	13.39	0.27%	0.28%
			4th gene			
0	5.34	0	5	0	6.89%	–
50	54.58	7.51	54.21	7.01	0.68%	7.04%
100	102.61	10.33	102.27	9.95	0.33%	3.81%
200	195.17	13.92	194.67	13.95	0.26%	0.25%
			5th gene			
0	50.66	0	50	0	1.32%	–
50	72.57	5.72	71.99	5.19	0.80%	10.24%
100	93.41	7.69	92.89	7.21	0.55%	6.58%
200	132.12	9.84	131.61	9.87	0.38%	0.31%

IV. DISCUSSION

In this article we presented the BSDE method to model simple gene expression networks. As a backward method, it relies on the specification of a gene network model parametrization and on endpoint conditions (as opposed to initial conditions). It can therefore be applied when we know, or can measure, the distribution of proteins at a given time, and we want to determine the distributions at previous time points. Another advantage of our method is that it allows to determine, and even to simulate if necessary, the trajectory of the noise process. To our knowledge, the noise process is usually unknown and cannot be determined by any forward method. In the BSDE method validation simulations, a good agreement was found between control and inferred protein level distributions, in terms of mean values and, in most cases, standard deviations. The BSDE method is therefore a powerful tool for time reversed simulations in gene networks / systems biology, where frequently an endpoint of interest is easily identifiable (and measured) and the aim is in assessing the prior (causal) conditions.

ACKNOWLEDGMENTS

R.C. was partially supported by FAPESP (Grant No. 2013/01242-8)

[1] M. Thattai and A. van Oudenaarden, Proc. Nat. Acad. Sci. **98**, 8614 (2001).
[2] E. Ozbudak, M. Thattai, I. Kurtser, A. Grossman, and A. van Oudenaarden, Nature Genetics **31**, 69 (2002).
[3] G. Karlebach and R. Shamir, Nat. Rev. Mol. Cell Biol. **9**, 770 (2008).
[4] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke, Biosystems **96**, 86 (2009).
[5] J. Mettetal, D. Muzzey, J. Pedraza, E. Ozbudak, and A. van Oudenaarden, Proc. Nat. Acad. Sci. **103**, 7304 (2006).
[6] D. Wilkinson, Bayesian Stat. **9**, 679 (2011).
[7] G. Lillacci and M. Khammash, Bioinformatics **29**, 2311 (2013).
[8] D. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[9] "COPASI: Biochemical System Simulator," `http://copasi.org/`.
[10] J. Bismut, J. Math. Anal. Appl. **44**, 384 (1973).
[11] E. Pardoux and S. Peng, Sys. Control Let. **14**, 55 (1990).
[12] J. Ma, P. Protter, and Y. J., Probability Theory and Related Fields **98**, 339 (1994).
[13] D. Gillespie, J. Chem. Phys. **113**, 297 (2000).
[14] K. Chen, T. Wang, H. Tseng, C. Huang, and C. Kao, Bioinformatics **21**, 2883 (2005).
[15] R. K. and D. J.H., Theor. Comp. Sci. **408**, 31 (2008).
[16] A. Kim, C. Martinez, J. Ionides, A. Ramos, M. Ludwig, N. Ogawa, D. Sharp, J. Reinitz, and M. Levine, PLoS Genetics **9** (2013), 10.1016/0022-247x(73)90066-8.
[17] A. V. Hill, J. Physiol. **40**, iv (1910)