Towards real-time control of gene expression at the single cell level: a stochastic control approach

Lakshmeesh R.M. Maruthi[1], Ilya Tkachev[1], Alfonso Carta[2], Eugenio Cinquemani[3], Pascal Hersen[4], Gregory Batt[5], and Alessandro Abate[6,1]

[1] Delft Center for Systems and Control, TU Delft, NL
[2] INRIA Sophia-Antipolis - Méditerranée, France
[3] INRIA Grenoble - Rhône-Alpes, France
[4] Laboratoire Matière et Systèmes Complexes, UMR 7057, Paris, France
[5] INRIA Paris-Rocquencourt, France
[6] Department of Computer Science, University of Oxford, UK
`gregory.batt@inria.fr alessandro.abate@cs.ox.ac.uk`

Abstract. Recent works have demonstrated the experimental feasibility of real-time gene expression control based on deterministic controllers. By taking control of the level of intracellular proteins, one can probe single-cell dynamics with unprecedented flexibility. However, single-cell dynamics are stochastic in nature, and a control framework explicitly accounting for this variability is presently lacking. Here we devise a stochastic control framework, based on Model Predictive Control, which fills this gap. Based on a stochastic modelling of the gene response dynamics, our approach combines a full state-feedback receding-horizon controller with a real-time estimation method that compensates for unobserved state variables. Using previously developed models of osmostress-inducible gene expression in yeast, we show *in silico* that our stochastic control approach outperforms deterministic control design in the regulation of single cells. The present new contribution leads to envision the application of the proposed framework to *wetlab* experiments on yeast.

1 Introduction

Gene expression plays a central role in the orchestration of cellular processes. The use of inducible promoters to change the expression level of a gene from its physiological level has significantly contributed to the understanding of the functioning of regulatory networks. Whereas the precise time-varying perturbation of the level of a target protein has the potential to be highly informative on the functioning of cellular processes, so far inducible promoters have been used for either static perturbations or simple dynamic perturbations with limited accuracy (see [14] for a notable exception). Alternative solutions, based on real-time control, have recently been proposed [11, 12, 16, 18]. In real-time, the level of the protein is observed and gene induction is modulated based on the distance to the objective. Thanks to the implementation of such external feedback loops, one can maintain the mean level of a fluorescent protein at some target value over extended time durations (set point experiments) and even follow time-varying profiles with good quantitative accuracy (tracking experiments). However, because of the significant cell-to-cell variability and the stochasticity of gene expression, even if the mean level of the protein follows precisely the objective, the performance of the controller is significantly worse when applied and measured at the single cell level. Yet if one wants to understand the effect of a perturbation of the level of a protein on a given process, one needs to control the level of this protein at the single cell level, that is, one needs to perform single cell control.

In [18] we have shown that single cell control is indeed effective: we have obtained better control performances when controlling single cells individually than when controlling the mean of the cell population. This slightly improved performance has been obtained by controlling the level of a particular, randomly-chosen cell using a deterministic model of gene expression. Given the stochasticity of cellular processes, one might wonder whether better control performances can be obtained by using a more appropriate stochastic model of gene expression. This question is actually not trivial: while the stochastic model is supposed to be closer to reality, it requires the use of complex controller architectures and the solution of computationally challenging optimization problems under tight time constraints.

In this work we investigate to what extent stochastic control techniques outperform more traditional deterministic control approaches. To do so, we consider a stochastic model of gene expression at the single cell level, alongside its deterministic counterpart, and develop state estimators and controllers for deterministic and stochastic control. We then compare the efficiency of the two approaches for set point regulation and tracking control in *in silico* experiments. Methodologically, in this work we introduce a stochastic receding horizon design approach of broad applicability, and a generalizable hybrid approach to state estimation. To our knowledge this is the first work on single cell control that accounts for gene expression noise.

The paper is structured as follows. In Section 2, we present the biological system, alongside the control platform used in [18] that has motivated this work, as well as the models used, inspired from [8, 21]. In Section 3 we present control algorithms for deterministic and stochastic control assuming full state observability, whereas in Section 4 we present a state estimation approach for stochastic models. The performances of deterministic and stochastic controllers are compared in Section 5 on two *in silico* control experiments.

2 Osmostress-induced gene expression in yeast

2.1 Hyper-osmotic stress response in yeast

In the budding yeast *S. cerevisiae*, an increase of the environmental osmolarity creates a water outflow and a cell shrinkage. The adaptation response to such an osmotic shock is mainly mediated by the high osmolarity glycerol (HOG) signal transduction pathway, leading to an increase of the cellular glycerol level via various mechanisms, one of which is the upregulation of genes involved in glycerol production. In [18], we have used the promoter of the osmoresponsive gene *STL1* to drive the expression of a yellow fluorescent protein, yECitrine, so as to monitor the gene expression response of the cells to repeated osmotic stresses (Fig. 1(a)).

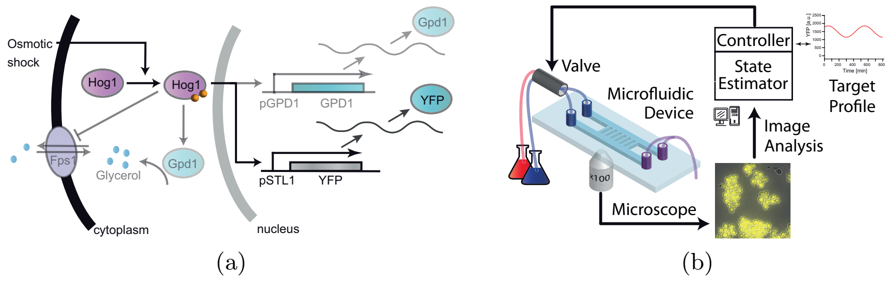

Fig. 1: The experimental setup. **(a)** Hyperosmotic shocks trigger the activation of the Hog1 protein and the intracellular accumulation of glycerol via short- and long-term adaptation responses (grayed). This system can be used to induce the production of a protein of interest, here a yellow fluorescent protein (YFP), by repeatedly applying hyperosmotic stresses. **(b)** Real-time control platform: single-cell and population control problems are defined respectively as controlling the fluorescence of a single randomly-chosen cell and the mean fluorescence of all the cells.

2.2 Platform for control of osmostress-induced gene expression

Using microfluidic devices one can grow yeast cells in monolayers over extended time durations. Because cells can be trapped in imaging chambers, their response can be tracked by fluorescence microscopy and their environment can be rapidly changed, thus enabling the repeated application of osmotic shocks (Fig. 1(b)). The addition of software for image analysis and for state estimation, and the computation of a control strategy closes the feedback loop. Experiments typically last 10-15 hours, with fluorescence measurements every 5-10 minutes.

2.3 Modeling osmostress-induced gene expression

We describe the osmostress induced gene expression by the reactions [21]

$$\begin{array}{lcl} pSTL1^{off} & \underset{c_2}{\overset{c_1 u}{\rightleftharpoons}} & pSTL1^{on} \\ pSTL1^{on} + CR & \underset{c_4}{\overset{c_3}{\rightleftharpoons}} & CR.pSTL1^{on} \\ CR.pSTL1^{on} & \xrightarrow{c_5} & CR.pSTL1^{on} + mRNA \\ mRNA & \xrightarrow{c_6} & mRNA + YFP \\ YFP & \xrightarrow{c_7} & \phi \\ mRNA & \xrightarrow{c_8} & \phi \end{array} \qquad (1)$$

Here $pSTL1^{off}$ and $pSTL1^{on}$ represent the inactive and the active states of the pSTL1 promoter, respectively. Furthermore, the interaction of $pSTL1^{on}$ with chromatin remodeling complexes (CR) enables the formation of the $CR.pSTL1^{on}$ complex and the effective transcription of mRNA, and the subsequent production of the fluorescent protein YFP. The degradations of the mRNA and the YFP protein follow first order kinetics. A change in the valve status from OFF to ON leads to an increase in the osmolarity of the cells environment, in the activation of the Hog1 protein, and in the increase of the effective input function u affecting promoter transition rates. The modeling of these processes is detailed in Appendix A.1, whereas the initial concentrations and the rate coefficients are listed in Table 2 in Appendix A.2.

A stochastic interpretation of the above reactions leads to a Chemical Master Equation (CME) model [6], characterized by a distribution accounting for the probability that the state of the system (represented by variables denoting molecular count) at time instant $t \in \mathbb{R}^+$ is $x(t)$, given its initial state $x(0)$ and an input signal $u(s), s \in [0, t]$. These stochastic semantics will be employed for testing the behavior of the model in *in silico* control experiments: in particular, we will use (a discrete-time version of) the Stochastic Simulation Algorithm (SSA) [6] to simulate the model. The dynamics can be approximated by a system of coupled deterministic dynamical equations, known as the Reaction Rate Equations (RRE) [6], operating over the concentrations x of the species as:

$$\dot{x}[i](t) = \sum_{j=1}^{M} v_{ij} a_j(x(t), u(t)), \qquad i = 1, \ldots, N. \tag{2}$$

Here the quantity M is the total number of reactions and N is the total number of species ($x[i]$ being the i^{th}). The vector $v_j := (v_{ij})_{i=1}^{N}$ is the state change vector for each reaction R_j: in particular v_{ij} represents the stoichiometry coefficients, defined as the change in the molecular population of a species S_i caused by the reaction R_j. Finally, the coefficients $a_j(\cdot)$ are the reaction rates, derived from the law of mass-action applied to (1): the control input in particular directly affects the affinity term a_1. The model in (2) is employed to synthetise a deterministic controller that will be used as a reference to assess the performances of the stochastic controller newly developed in this work. For the latter objective, a second approximation of the CME dynamics is introduced in Section 4, in order to derive an efficient state estimation scheme developed in the context of noisy partial observations, which combines the original CME semantics with a Chemical Langevin Equation (CLE) approximation.

3 Single-cell control with full state information

The control of gene expression is treated as a model-based optimal control problem. The goal of the control synthesis problem is to track a given profile of protein concentration over a finite time horizon T. As in [18], we require that the controller complies with particular timing constraints: the valve should remain ON at least 5 minutes and at most 8 minutes, and two stress inputs must be separated by at least 20 minutes (see Appendix A.1 for more details). These constraints are imposed in order to prevent cell adaptation to hyperosmotic environments.

In this section, the availability of full-state information (namely, knowledge of the values of *all the variables*) is assumed. Above we have formulated two models: a stochastic discrete-state one and a deterministic continuous-state one. For both cases, a control synthesis architecture based on the classical dynamic programming (DP) paradigm is proposed. As the classical DP suffers from the curse of dimensionality, we employ an approximate DP method called Fitted Q-Iteration (FQI) [4, 9], tailored here to the finite-horizon setting. The FQI algorithm applies the idea of value iteration to the so-called Q-functions: a Q-function approximation is used in place of a value function approximation, and it allows for an immediate computation of the optimal actions at each optimisation stage. The FQI algorithm offers the possibility to employ powerful regression algorithms from supervised learning to interpolate the Q-function computed over a finite set of states to cover the entire state space [9].

Optimal controller synthesis via DP For the controller synthesis problem, we will adopt a discrete time simulation framework. Let us denote the state space by X, the action space by U, and the space supporting the noise term by W. For each $x \in X$ we denote by $U(x) \subseteq U$ the set of actions enabled at x. A stochastic discrete-time dynamical system is described by the following difference equation:

$$x_{k+1} = f(x_k, u_k, w_k), \qquad k = 1, \dots, T-1, \tag{3}$$

where $x_k \in X$ is the state of the system at time k, $u_k \in U(x_k)$ is the action taken at time k, and $w_k \in W$ is the noise variable with a specified distribution: let us remark that the recursive dynamics in (3) can be equivalently expressed by a conditional distribution $x_{k+1} \sim P(\cdot|x_k, u_k)$ [10], which in our instance can be derived from a discrete-time version of the CME that we have discussed in the previous section.

A control policy is a sequential decision rule $\pi = (\pi_k)_{k=0}^{T-1}$, where $\pi_k : X \to U$ has to be chosen over admissible controls only: $\pi_k(x) \in U(x)$ for all $x \in X$. The instantaneous cost $c_k(x_k, u_k)$ is comprised within an (expected) additive performance criterion over a finite time horizon, which for a fixed policy π is given by

$$Q_0^\pi(x_0, u_0) := \mathbb{E}\left[c_T(x_T) + \sum_{k=0}^{T-1} c_k(x_k, \pi_k(x_k))\right]. \tag{4}$$

Notice that the terminal cost, c_T, depends only on the state variable. In the following, we shall employ the cost function $c_k(x_k, u_k) = |\mathit{YFP}_k - \mathit{YFP}_{ref,k}|$, which penalises deviations from the reference profile $\mathit{YFP}_{ref,k}$, and a null terminal cost c_T. We are interested in the policy π^* that minimizes the cost:

$$Q_0^*(x, u) := \inf_\pi Q_0^\pi(x, u) = Q_0^{\pi^*}(x, u).$$

This cost can be obtained via DP by the backward recursion, initialised at the value c_T and propagated as:

$$Q_k^*(x,u) = \mathcal{T} Q_{k+1}^*(x,u), \tag{5}$$

where $\mathcal{T}$ is an operator acting on functions $H : X \times U \to \mathbb{R}$ as follows:

$$\mathcal{T} H(x,u) := c(x,u) + \inf_{u' \in U} \mathbb{E} H(f(x,u,w),u'). \tag{6}$$

An optimal policy can be computed as

$$\pi_k^*(x) \in \arg\min_{u \in U} Q_{k+1}^*(x,u), \qquad k = 0, \ldots, T-1. \tag{7}$$

The Q-iteration in (5)-(7) is computationally unfeasible for problems with extended state spaces, and in particular with the single-cell control problem we are dealing with: we approximate its solution by means of a stochastic FQI [9].

FQI for the stochastic model The FQI is a batch-mode algorithm computed offline, which fits an approximation architecture to the Q-function defined over $X \times U$ using a set of tuples

$$\mathcal{F} = (x^i, u^i, c^{ij}, z^{ij}), \qquad i = \{1, \ldots, m_x\},\ j = \{1, \ldots, m_z\}, \tag{8}$$

where $x^i \in X$ is the instance of the current (or reference) state, $u^i \in U(x^i)$ is the corresponding action, $z^{ij} \in X$ is a possible successor state under the action u^i, c^{ij} is the cost associated with a transition of the state from x^i to z^{ij}, m_x is the number of current states, m_z is the number of successor states that are needed for the evaluation of the expectation operator in (6) using Monte-Carlo integration.

We adopt an offline approach, owing to the computational complexity of the optimisation problem and to the stringent online time requirements. Using the batch of samples in (8), Algorithm 1 (in the Appendix) computes an approximation of the Q-function through a backward recursion from time instant T to 1. Each iteration of the algorithm consists of the following two steps:

- In the first step, the backward recursion for the Q-function at time $k+1$ is evaluated using a Monte-Carlo integration. The operator $\mathcal{T}$ is approximated by an empirical operation $\hat{\mathcal{T}}_{\mathcal{F}}$ as shortly defined in (9): namely the value of $\mathcal{T}\hat{Q}_{k+1}$ is estimated as $\hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{k+1}$, for all x^i, $i = 1, \ldots, m_x$.
- The second step involves fitting the approximation function $\hat{Q}_k$ to $\hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{k+1}$: the optimal fit $\hat{Q}_k$ is achieved by means of a regression algorithm.

The overall performance and computational complexity of the FQI method heavily hinges on the choice of the regression algorithm. The supervised learning paradigm offers a wide range of algorithms that can be used for regression [3]. We have made use of the Fixed-Size Least-Squares Support Vector Machine (LS-SVM) [5], due to its computational efficiency and its powerful capability of generalisation. The LS-SVM model provides two parameters for tuning: the squared bandwidth σ^2 and the regularization parameter γ, which have here been tuned manually through trial and error (but could be as well be optimised over with a more sophisticated alternative). These parameters are crucial to determine the trade-off between the training error minimization, the smoothness and the generalization. Algorithm 1 is detailed in the Appendix: there, we assume that the regression algorithm is fixed, and denote by $\mathcal{G}$ the corresponding space of test functions $G : X \times U \to \mathbb{R}$. For a given tuple $\mathcal{F}$ we denote

$$\hat{\mathcal{T}}_{\mathcal{F}} H(x^i, u^i) := \inf_{u' \in U(x^i)} \frac{1}{m_z} \sum_{j=1}^{m_z} \left[c^{ij} + H(z^{ij}, u') \right] \tag{9}$$

and the corresponding 2-norm as $\|H' - H''\|_{\mathcal{F}} := \sum_{i=1}^{m_x} \left| H'(x^i, u^i) - H''(x^i, u^i) \right|^2$.

FQI for the deterministic model A discrete-time deterministic model is a special case of (3) where the update law f does not depend on the noise variable w. In our work, we refer to the deterministic dynamics discussed in (2), after time discretization. For this simpler setup, the DP operator takes the form

$$\mathcal{T} H(x, u) = c(x, u) + \inf_{u' \in U} H(f(x, u), u'),$$

and no expectation evaluations are needed. Thus, we have $m_z = 1$, so that only one successor state is needed for each instance of the state. As a result,

$$\hat{\mathcal{T}}_{\mathcal{F}} H(x^i, u^i) = \inf_{u' \in U(x^i)} \left[c^i + H(z^i, u') \right].$$

One can therefore directly tailor Algorithm 1 to the deterministic case.

Practical implementation of the stochastic FQI via receding horizon strategy Although the FQI for the deterministic model works well within our setup, the FQI algorithm for the stochastic model over the entire experimental duration (denoted by the time horizon T) has been found to be computationally infeasible, since parameters achieving a good generalisation for the regression algorithm over the complete time horizon T are not easily found, and because of the Monte-Carlo computations that are instead absent in the deterministic case. In order to overcome this issue, we have embedded the FQI algorithm into a receding horizon strategy, resulting in a stochastic receding horizon scheme (see Algorithm 2 in the Appendix) [1]. In short, over a finite prediction horizon $T_p \ll T$, the Q-functions are approximated offline using Algorithm 1. After the computation of the optimal control sequence and the application of the current control action, the horizon is shifted by one sample and the optimisation is performed again, until the whole horizon T is covered.

4 Partial information case: estimation of system states

Typically not all state variables of a biological model are observed directly. This is in particular the case for the yeast osmotic shock response system, where only protein levels are observable via noisy measurements:

$$y_k = \mathit{YFP}_k + e_k, \quad e_k = (e_a + e_b \cdot \mathit{YFP}_k)\eta_k, \tag{10}$$

where for $k = 1, \ldots, T$, y_k is the measurement at time k for a given cell, and η_k are i.i.d. standard normal random variables, whereas e_a and e_b are the intensity of the additive and multiplicative parts of the measurement noise.

In practice, the state-feedback control must rely on estimates of the state that are generated online from the available measurements. Here we develop a strategy for real-time state estimation with reference to yeast osmotic shock response. We observe that the strategy can be applied to other biological scenarios.

We start from the continuous-time stochastic Markov model of the CME, which is expressed in terms of discrete-valued state variables x. One possible approach for estimating state x from measurements y_k is particle filtering [2]. In particle filtering, N hypothetical evolutions of the system state are randomly simulated up to the next measurement. When the latter becomes available, state estimates are produced by weighting the simulated trajectories, where the weights quantify the relevance of every simulated trajectory to the new (partial) state measurement. Since particles have to explore a large (possibly infinite) state space, in practice particle filtering requires many (e.g. $N > 1000$) simulations of the system, which makes it poorly suited for online applications. In [2], we have proposed an alternative approach using Unscented Kalman Filtering (UKF) [19] and based on the CLE, a continuous-valued approximation of the CME model [7]. In the current context, this approach is partly inappropriate, since the promoter state variables are inherently discrete (they take values 0 or 1 only). In order to combine the flexibility of particle filtering with the computational advantages of UKF, we propose to limit the Langevin approximation to the mRNA and protein dynamics.

We first note that promoter dynamics do not depend on mRNA and protein abundance. Let us partition the state variables as $x = (x^d, x^c)$, where

$$x^d = (\mathit{pSTL1}^{\mathit{off}}, \mathit{pSTL1}^{\mathit{on}}, \mathit{CR} \cdot \mathit{pSTL1}^{\mathit{on}}), \qquad x^c = (\mathit{mRNA}, \mathit{YFP}).$$

Consider a model where the dynamics of x^d (not depending on x^c) are left unchanged (*i.e.*, follow the original CME), while for any given trajectory of x^d, the dynamics of x^c are approximated by the Langevin equation

$$dx^c[i] = \sum_{j=1}^{M} v^c_{ij} a_j(x^c, x^d)dt + \sum_{j=1}^{M} v^c_{ij} \sqrt{a_j(x^c, x^d)}dW_j, \qquad i = 1, 2. \tag{11}$$

Here, for $j = 1, \ldots, M$, W_j are independent Wiener processes and $v^c_{\cdot j}$ is the subvector of v_j corresponding to x^c. The relevance of the Langevin approximation to mRNA and protein dynamics has been discussed in [7] and, for filtering applications, it has been assessed on a different but relevant system in [2]. Note that, while x^d remains discrete-valued, x^c may now take continuous values.

Based on this hybrid model, a filtering procedure iterating over subsequent measurement indices k combining importance (particle) filtering with UKF is obtained as follows. At time t_{k-1}, let $\hat{x}^c_{k-1|k-1}$ be the estimate of the current state

x^c based on measurements $y_0, \dots, y_{k-1}$, and let $\hat{x}^{d,i}_{k-1|k-1}$, with $i = 1, \dots, N$, be N putative values of the current state x_d (with N small, see below). For every i, a hypothetical discrete-state trajectory $\hat{x}^{d,i}_{k-1}(t)$, with $t \in [t_{k-1}, t_k)$, is generated by stochastic simulation of the discrete-state dynamics starting from $\hat{x}^{d,i}_{k-1|k-1}$. Over the same time horizon, for every i, mRNA and protein state predictions $\hat{x}^{c,i}_{k-1}(t)$ are computed along trajectory $\hat{x}^{d,i}_{k-1}(t)$ via UKF. When the next protein measurement y_k becomes available, based on measurement model (10), an importance weight w_i, proportional to the likelihood of y_k given the hypothetical state value $\hat{x}^{c,i}_{k-1}(t_k)$, is computed for every particle i. Note that weights w_i play the role of a-posteriori probabilities of the different particles. Also, continuous-state predictions $\hat{x}^{c,i}_{k-1}(t_k)$ are updated to estimates $\hat{x}^{c,i}_{k|k}$ of the current state x^c by integrating the new piece of information provided by $y(t_k)$, in accordance with the so-called measurement-update step of UKF. At this stage, the ensemble (Conditional Expectation) estimate $\hat{x}^c_{k|k}$ as well as an ensemble (Maximum-A-Posteriori) estimate $\hat{x}^d_{k|k}$ for the discrete state are computed as

$$\hat{x}^d_{k|k} = \arg \max_{z \in \{0,1\}^3} \sum_i 1_z\left(\hat{x}^{d,i}_{k-1}(t_k)\right) \cdot w_i, \qquad \hat{x}^c_{k|k} = \sum_i \hat{x}^{c,i}_{k|k} \cdot w_i, \tag{12}$$

where $1_z(\cdot)$ is the indicator function. For control purposes, these are the estimates that are passed to the controller with entries of $\hat{x}^c_{k|k}$ rounded to the nearest integers. To proceed for the next iteration of the algorithm, the new putative values of the discrete state $\hat{x}^{d,j}_{k|k}$, with $j = 1, \dots, N$, are set equal to the result of N independent random extractions from the pool of particles $\{\hat{x}^{d,i}_{k-1}(t_k)\}_{i=1,\dots,N}$, with sampling probabilities equal to w_i (resampling step of particle filtering). The whole procedure is summarized in Algorithm 3 in the Appendix.

The initialization of the procedure at the starting time $k = 0$ is performed based on the a priori statistics of x^d and x^c. Given the small (finite) discrete state space of x^d, a number of particles N much smaller than traditional particle filter implementations is expected to suffice. Empirical evaluation (not reported here) has led to select $N = 50$, a value above which no significant improvement of filtering performance has been observed. The implementation of the UKF procedure is analogous to that of [2] and is omitted for brevity. We just note that, at every step k and for every particle i, UKF requires the numerical solution of $2n^c + 1$ ODEs over the time span $[t_{k-1}, t_k)$, with $n^c = 2$ being the number of continuous states. The solution of these ODEs can be carried on in parallel with the simulation of $\hat{x}^{d,i}_{k-1}$. Contrary to the control module, resorting to time discretization is not needed, although it can be considered towards higher computational efficiency.

5 Results

5.1 Deterministic and stochastic control in the full information case

In this section we present the results of the control of gene expression to track time-homogeneous and time-varying target profiles, using the deterministic and stochastic controllers detailed in Section 3. In order to test the effectiveness of the proposed algorithms, the controller trained using the deterministic FQI was first tested over the deterministic RRE model. As expected in this case, the controller has successfully been able to track the signals (see Appendix A.4 for implementation details). To test the control performance in a realistic biological context, this controller has then been used over the stochastic CME model. At the maximum, the controller is able to track the reference signal to within a deviation of 10% as shown in Fig. 2(a)(b). The deterministic controller has then been replaced with the stochastic controller (see Appendix A.3 for implementation details) and it has been found that the stochastic controller is able to track the reference signal to within a deviation of 5% from the reference trajectory (Fig. 2(d)(e)).

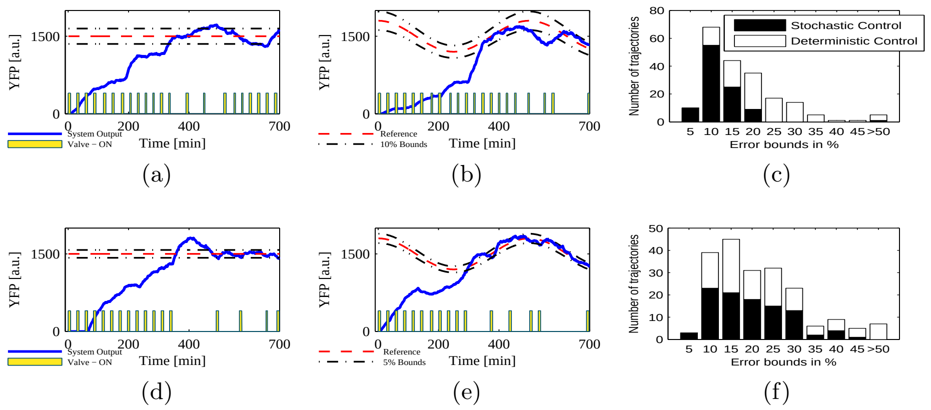

Fig. 2: Comparison of stochastic and deterministic control schemes in the full information case, run over the probabilistic model. (a)(b) Deterministic controller tracking the desired profiles with a shown deviation of 10% from reference trajectories. (d)(e) Stochastic controller showing improved performance with a deviation of 5%. (c)(f) Monte-Carlo simulations validating the superior performance of the stochastic controller over its deterministic counterpart: the histogram plots the number of closed-loop trajectories falling within specific error bounds from the reference trajectory.

In order to get a quantitative comparison of the performance of the stochastic controller over the deterministic controller, 100 runs of each algorithm have been performed using Monte-Carlo simulations. To measure the quality of the control, we have used $\epsilon := \frac{1}{T-T_0}\sum_{k=T_0}^{T} |YFP_k - YFP_{ref,k}| / YFP_{ref,k}$, where T_0 is the time it takes the system to reach the desired trajectory. In practice, we have chosen $T_0 = 400$ and $T_0 = 300$ minutes for the set point and signal tracking experiments, respectively. These results are presented in Fig. 2(c)(f). It is evident from the figure that the controller developed considering the stochastic nature of the gene expression yields superior performance than the controller developed ignoring it.

5.2 Stochastic control with partial information

The control laws obtained in the full information case are functions of the current state x_k: at each time k it is supposed that the controller observes the exact value of the full current state x_k and that it applies the appropriate action. In reality the measurements y_k are limited to the fluorescent protein. The hybrid filter detailed in Section 4 has been used to extract information about the states of the gene expression network using 50 particles. The filter does not succeed to accurately track the switching of the discrete states of the promoter but is able to track the *mRNA* and *YFP* protein concentrations fairly accurately (Fig. 3(a)). The filter has then been used in conjunction with the stochastic controller: the simulation results presented in Figure 3(b)(c) show that the controller is robust to state estimation errors and is able to successfully track the reference profiles.

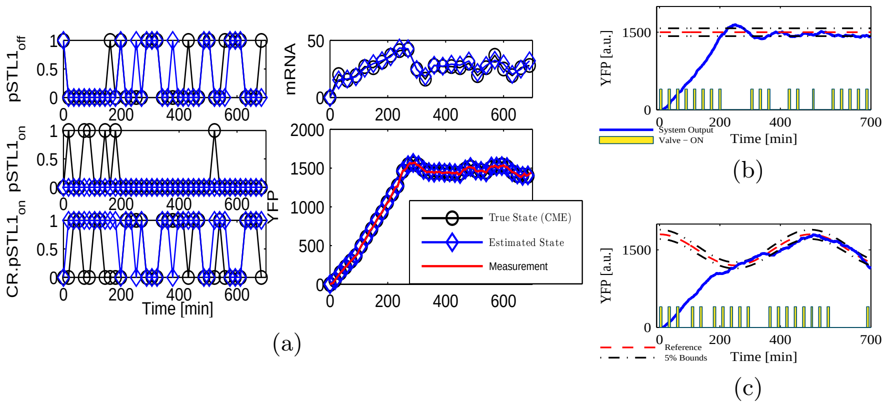

Fig. 3: Results of the stochastic control scheme run with the hybrid filter in the partial information case. (a) State estimation shows accurate results for mRNA and YFP, whereas the filter faces difficulties estimating the switching action of the promoter. (b)(c) Controller robustness over state estimation errors and ability to track reference signals to within a deviation of 5%.

6 Discussion and conclusions

The main contribution of this paper is the development of a complete model-based control framework adapted to stochastic models of gene expression. Although the identification of stochastic models of gene expression has recently been extensively studied, the control of gene expression using stochastic models has been barely addressed so far. This goal requires the non-trivial development of integrated stochastic state estimators and controllers. We have demonstrated *in silico* that stochastic control has the potential to deliver superior performances in comparison to a deterministic counterpart explored in earlier literature. This work paves the way for the development of an experimental platform for single-cell control based on optogenetics solutions, which enable the independent stimulation of live single cells in real-time [16, 20].

Acknowledgments The first two authors are students and the main contributors to this work. This research has been supported by a van Gogh grant from the FR-NL Academie van Wetenschappen, by the European Commission IAPP project AMBI 324432, by the GeMCo (ANR-10-BLAN-0201) and ICEBERG (ANR-10-BINF-06-01) grants from the Agence Nationale de la Recherche, and by the Action d'Envergure ColAge from INRIA/INSERM.

References

1. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
2. A. Carta and E. Cinquemani. State estimation for gene networks with intrinsic and extrinsic noise: a case study on *E.coli* arabinose uptake dynamics. In *European Control Conference, ECC'13*, Zurich, Suisse, 2013.
3. R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proc. of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.
4. D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, pages 503–556, 2005.
5. M. Espinoza, J.A.K. Suykens, and B. De Moor. Fixed-size least squares support vector machines: A large scale application in electrical load forecasting. *Computational Management Science*, 3(2):113–129, 2006.
6. D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.
7. D.T. Gillespie. The chemical Langevin equation. *Journal of Chemical Physics*, 113(1):297–306, 2000.
8. A.M. Gonzalez, J. Uhlendorf, E. Cinquemani, G. Batt, and G. Ferrari-Trecate. Identification of biological models from single-cell data : A comparison between mixed-effects and moment-based inference. In *European Control Conference, ECC'13*, pages 3652–3657, 2013.
9. S. Haesaert, R. Babuska, and A. Abate. Sampling-based approximations with quantitative performance for the probabilistic reach-avoid problem over general Markov processes. *arXiv preprint*, 2014. arXiv:1409.0553.
10. O. Kallenberg. *Foundations of modern probability*. Probability and its Applications. Springer Verlag, New York, 2002.
11. F. Menolascina, G. Fiore, E. Orabona, L. De Stefano, M. Ferry, J. Hasty, M. di Bernardo, and D. di Bernardo. In-vivo real-time control of protein expression from endogenous and synthetic gene networks. *PLoS Computational Biology*, 10(5):e1003625, 2014.
12. A. Milias-Argeitis, S. Summers, J. Stewart-Ornstein, I. Zuleta, D. Pincus, H. El-Samad, M. Khammash, and J. Lygeros. In silico feedback for in vivo regulation of a gene expression circuit. *Nature Biotechnology*, 29:1114–1116, 2011.
13. D. Muzzey, C.A. Gómez-Uribe, J.T. Mettetal, and A. van Oudenaarden. A systems-level analysis of perfect adaptation in yeast osmoregulation. *Cell*, 138(1):160–171, 2009.

14. E.J. Olson, L.L. Hartsough, B.P. Landry, R. Shroff, and J.J. Tabor. Characterizing bacterial gene circuit dynamics with optically programmed gene expression signals. *Nature Methods*, 11:449–455, 2014.
15. K. Pelckmans, J.A.K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, B. De Moor, and J. Vandewalle. LS-SVMlab: a matlab/c toolbox for least squares support vector machines. *Tutorial. Leuven, Belgium*, 2002.
16. J.E. Toettcher, D. Gong, W.A. Lim, and O.D. Weiner. Light-based feedback for controlling intracellular signaling dynamics. *Nature Methods*, 8:837–839, 2011.
17. J. Uhlendorf, S. Bottani, F. Fages, P. Hersen, and G. Batt. Towards real-time control of gene expression: controlling the HOG signaling cascade. In *16th Pacific Symposium of Biocomputing*, pages 338–349, 2011.
18. J. Uhlendorf, A. Miermont, T. Delaveau, G. Charvin, F. Fages, S. Bottani, G. Batt, and P. Hersen. Long-term model predictive control of gene expression at the population and single-cell levels. *PNAS*, 109(35):14271–6, 2012.
19. E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC'00*, pages 153–158. IEEE, 2000.
20. X. Yang, A. Payne-Tobin Jost, O.D. Weiner, and C. Tang. A light-inducible organelle-targeting system for dynamically activating and inactivating signaling in budding yeast. *Molecular biology of the cell*, 24(15):2419–30, 2013.
21. C. Zechner, J. Ruess, P. Krenn, S. Pelet, M. Peter, J. Lygeros, and H. Koeppl. Moment-based inference predicts bimodality in transient gene expression. *PNAS*, 109(21):8340–8345, 2012.

A Appendix

A.1 Implementation constraints on the control of gene expression

In order to limit cell adaptation to hyperosmotic environments, we delimit the duration of hyperosmotic shocks to 8 minutes and impose at least a 20 minute time lag between two successive shocks. We also require that shocks last at minimum 5 minutes.

As shown in [18], there is a known lag between the valve actuation and the actual change of osmolarity of the cellular environment in the imaging chamber. Formally, for a given osmotic shock, we denote by t_{on} and t_{off} the times at which the valve switches to ON and to OFF positions, respectively, and represent the osmolarity h in the imaging chamber as follows (see Figure 4).

$$h(t) = \begin{cases} 0 & \text{if } t < t_{on} + 2, \\ t - (t_{on} + 2) & \text{if } t_{on} + 2 < t < t_{on} + 3, \\ 1 & \text{if } t_{on} + 3 < t < t_{off} + 2, \\ 1 - (t - (t_{off} + 2))/4 & \text{if } t_{off} + 2 < t < t_{off} + 6, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

As in [8, 13, 17], we assume that the activity s of the Hog1 protein depends on the osmolarity of the environment h as follows.

$$\dot{s}(t) = \kappa h(t) - \mathit{\Gamma} s(t), \tag{14}$$

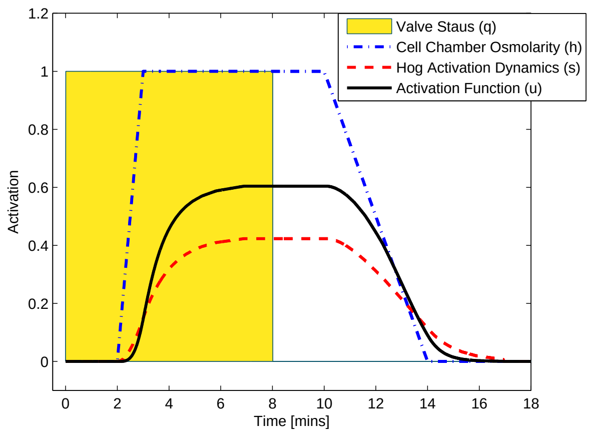

Fig. 4: Temporal evolution of the osmolarity of the cellular environment h, of the Hog1 activity s, and of the promoter activation stochastic rate u, as a function of the position of the microfluidic valve (0/1: normal/hyper-osmotic medium).

with $s(0) = 0$; we further assume that the *pSTL1* promoter activation stochastic rate u is a function of the Hog1 activity s, following Hill-type kinetics as

$$u(t) = \frac{(s(t) + a_0)^{n_H}}{{K_d}^{n_H} + (s(t) + a_0)^{n_H}}. \tag{15}$$

Note that we assume here that there is no significant stochasticity in signal transduction. The rate parameters used for model simulation are listed in the table below. Also in practice, because the controller uses a discrete time representation, we refer to the input at instant k as $u_k = u(t_k)$.

Parameter	Value	Parameter	Value
κ	0.3968 (a.u.)	Γ	0.9225 (a.u.)
K_d	0.34906 (a.u.)	a_0	0.0027998 (a.u.)
n_H	2.1199 (a.u.)		

Table 1: Rates of the activation function.

A.2 Parameters employed in the simulation and analysis of the model

The rate parameters and initial concentrations used for the simulation of the model are listed in the two tables below.

A.3 Implementation details of the FQI algorithm over the stochastic CME model

For the stochastic receding horizon control approach, the samples x^i have been drawn corresponding to a single system trajectory. The trajectory has been generated by simulating the system using a discrete time version of the stochastic

Parameter	Value	Parameter	Value
$(pSTL11^{off})_0$	1 (a.u.)	$(pSTL11^{on})_0$	0 (a.u.)
CR_0	102.51 (a.u.)	$(CR \cdot pSTL11^{on})_0$	0 (a.u.)
$mRNA_0$	0 (a.u.)	YFP_0	0 (a.u.)
c_1	23.604 $(min)^{-1}$	c_5	12.256 $(min)^{-1}$
c_2	180.03 $(min)^{-1}$	c_6	0.36113 $(min)^{-1}$
c_3	0.024559 $(min)^{-1}$	c_7	0.025091 $(min)^{-1}$
c_4	0.9384 $(min)^{-1}$	c_8	0.003354 $(min)^{-1}$

Table 2: Initial concentrations and rates of the stochastic gene expression network.

Parameter	Value	Parameter	Value
e_a	1.0115 $(min)^{-1}$	e_b	0.0037 $(min)^{-1}$

Table 3: Parameters of the measurement model.

simulation algorithm. The intrinsic variability results from the stochasticity of the CME.

For each x^i, 250 tuples ($m_x = 250$) of the form (x^i, u^i) have been generated. For each tuple, the system has been simulated 100 times ($m_z = 100$) to obtain the next state z^{ij} to evaluate the Monte-Carlo integration. The cost c^{ij} has been computed as explained in main text and a single batch of 25000 tuples ($\mathcal{F}_s = 25000$) has been obtained. The optimization has been performed for a prediction horizon T_p of 8 minutes and for a time horizon T of 700 minutes. The discretization interval Δt has been set to 0.008 min. The squared bandwidth σ^2 and the regularization parameter γ of the regression algorithm have been tuned by a trial and error method and the final parameters have been reported below.

Squared Bandwidth (σ^2)	Regularization Parameter (γ)
600000	500

Table 4: Tuned LS-SVM parameters to track time varying and time constant profiles using the controller trained on the stochastic CME model.

The stochastic FQI and receding horizon algorithms respectively are detailed below.

Algorithm 1 Stochastic Finite Horizon FQI (T, $\mathcal{F}$)

```
1: Initialize the parameters of the regression algorithm $\sigma^2$ and $\gamma$ and set $\hat{Q}_T$ to 0
2: for $k := T - 1$ to 0 do
3:     Estimate $\hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{k+1}$.
4:     Find the fit that minimizes the 2-norm loss by means of a regression algorithm
           $$\hat{Q}_k(x,u) = \arg\min_{G \in \mathcal{G}} \|G - \hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{k+1}\|_{\mathcal{F}}.$$
5: end for
```

Algorithm 2 Stochastic Receding Horizon Control (T, T_p, $\mathcal{F}$)

```
1: for k:=1 to $T$ do
2:     Initialize the parameters of the regression algorithm $\sigma^2$ and $\gamma$ and set $Q_T$ to 0
3:     for $l := k + T_p$ to $k$ do
4:         Estimate $\hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{l+1}$.
5:         Find the fit minimizing the 2-norm loss by means of a regression algorithm
               $\hat{Q}_l(x,u) = \arg\min_{G\in\mathcal{G}} \|G - \hat{\mathcal{T}}_{\mathcal{F}}\hat{Q}_{l+1}\|_{\mathcal{F}}.$
6:     end for
7: end for
```

A.4 Implementation details of the FQI algorithm over the deterministic RRE model

For the deterministic control approach presented in Section 3, 400 tuples have been generated corresponding to a single trajectory. The trajectory has been obtained by simulating reactions of the gene expression network using the RRE. A time horizon of 700 minutes has been considered and the regression algorithm has been implemented using the LS-SVM MATLAB toolbox in [15]. The Fixed-Size LS-SVM model provides two parameters for tuning: the squared bandwidth σ^2 and the regularization parameter γ. The parameters have been tuned manually using a trial-and-error method, and the selected ones are reported in Tables 5 and 6 below.

Time Horizon (T)	Squared Bandwith (σ^2)	Regularization Parameter (γ)
700 - 651	40000	10^{-1}
650 - 601	40000	10^{-2}
600 - 551	40000	10
550 - 501	40000	200
500 - 451	40000	1
450 - 401	40000	200
400 - 351	40000	100
350 - 301	40000	200
300 - 251	40000	100
250 - 201	40000	300
200 - 151	40000	100
150 - 1	40000	100

Table 5: Tuned LS-SVM parameters to track a set-point of 1500 (a.u.) using the controller trained on the deterministic RRE model.

For the deterministic control approach, the deterministic version of the FQI algorithm has been trained and implemented over the RRE model. The simulation results in Figure 5 show that the system is able to track the reference profiles within a maximum deviation of 5%.

Time Horizon (T)	Squared Bandwith (σ^2)	Regularization Parameter (γ)
700 - 651	400000	50
650 - 601	40000	9
600 - 551	20000	1
550 - 501	400000	30
500 - 451	40000	596
450 - 401	40000	800
400 - 351	40000	300
350 - 301	100000	1
300 - 251	100000	11
250 - 201	100000	5
200 - 1	100000	4000

Table 6: Tuned LS-SVM parameters to track the sinusoidal reference signal using the controller trained on the deterministic RRE model.

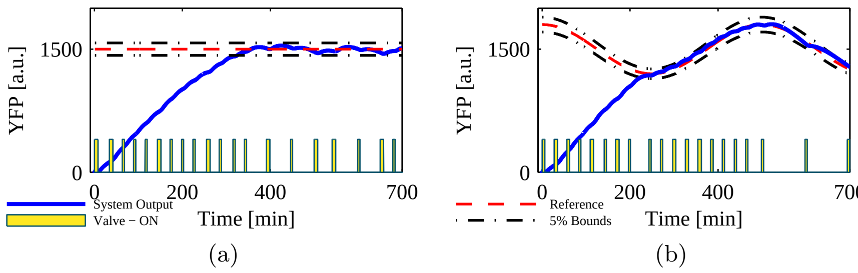

Fig. 5: Results for the deterministic control scheme in the full information case. The deterministic controller tracks time-varying and constant profiles within a deviation of 5% from the reference trajectory.

A.5 Hybrid Estimation Algorithm

Algorithm 3 Hybrid Filter for Estimation of the Model States

```
1: Initialize $\hat{x}^{d,i}_{0|-1}(0)$, $\hat{x}^{i,c}_{0|-1}(0)$, and $w_i$, with $i = 1, \dots, N$, s.t. $\sum_i w_i = 1$
2: for $k = 0, 1, 2, \dots$ do
3:     Acquire new measurement $y_k$
4:     Compute and normalize weights $w_i \propto \log p\big(y_k | \hat{x}^{c,i}_{k-1}(t_k)\big)$, with $i = 1, \dots, N$
5:     Compute UKF estimate $\hat{x}^{i,c}_{k|k}$ from $\hat{x}^{i,c}_{k-1}(t_k)$ and $y_k$, with $i = 1, \dots, N$
6:     Compute and provide ensemble estimates (12)
7:     Define $N$ new particles $\hat{x}^{i,d}_{k|k}$ by resampling particles $\{\hat{x}^{i,d}_{k-1}(t_k)\}$ with prob. $\{w_i\}$
8:     Simulate $\hat{x}^{i,d}_{k}(t)$, $t \in [t_k, t_{k+1})$, from $\hat{x}^{i,d}_{k}(t_k) = \hat{x}^{i,d}_{k|k}$, with $i = 1, \dots, N$
9:     Compute UKF prediction $\hat{x}^{i,c}_{k}(t)$ along $\hat{x}^{i,d}_{k}(t)$, $t \in [t_k, t_{k+1})$, with $i = 1, \dots, N$
10: end for
```