

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

Lakshmeesh R.M. Maruthi¹, Ilya Tkachev¹, Alfonso Carta², Eugenio Cinquemani³, Pascal Hersen⁴, Gregory Batt⁵, 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 and alessandro.abate@cs.ox.ac.uk

Abstract. Recent works have demonstrated the experimental feasibility of real-time gene expression control based on deterministic controller designs. By taking control of the level of intracellular proteins, one can probe the cell dynamics with unprecedented flexibility. However, single-cell dynamics are stochastic in nature, and a control framework explicitly accounting for this variability is lacking. In this work we devise a stochastic Model Predictive Control (MPC) framework that fills this gap. Based on stochastic modelling of 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 may outperform deterministic control design in the control of single cells. Application of the proposed framework to real experiments on yeast is envisioned.

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 [12] for a notable exception). Alternative solutions, based on real-time control, have recently been proposed [11, 14, 10, 15]. 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 (setpoint experiments) and even obtain 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 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 [15] we have shown that single cell control is indeed effective: we obtained better control performances when controlling single cells individually than when controlling the mean of the cell population. This slightly improved performance was 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 then 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 necessitates the use of complex controller architectures and the resolution of computationally challenging optimization problems under tight time constraints.

Here, 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 setpoint regulation and tracking control *in silico* experiments. Methodologically, we have introduced a stochastic MPC 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 and the control platform used in [15] that has motivated this work, as well as the models used, inspired from [18, 8]. 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 being the upregulation of genes involved in glycerol production [9]. In [15], we used the promoter of the osmosensitive 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)).

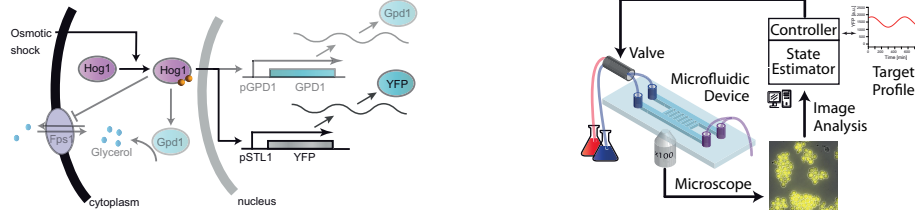


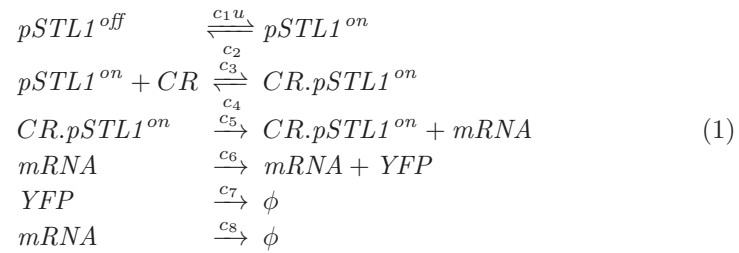
Fig. 1: The experimental setup. **(a)** Hyperosmotic shocks trigger the activation of the Hog1 protein and the intracellular accumulation of glycerol via short- and long-term adaptation responses (grayed). This system can be used to induce the production of a protein of interest, here a yellow fluorescent 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 10-20 hours, with fluorescence measurements every 5-10 minutes.

2.3 Modeling osmostress-induced gene expression

As in [18], we represent the osmostress induced gene expression by the following set of reactions.



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 degradation 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. The initial concentrations, the rate coefficients, and the parameters of the activation dynamics are listed in Table 4 in the Appendix.

The stochastic interpretation of the above reactions leads to a Chemical Master Equation (CME) model, which fully characterizes the quantity $P(x(t)|x(0))$, namely the probability that the global state of the system (represented by variables denoting molecules count) at time instant $t \in \mathbb{R}^+$ is $x(t)$, given its initial state [6].

The dynamics can be alternatively 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:

$$\dot{x}_i(t) = \sum_{j=1}^M v_{ij} a_j(x(t)), \quad i = 1, \dots, N. \quad (2)$$

Here the quantity M is the total number of reactions and N is the total number of species. The vector $v_j := (v_{ij})_{i=1}^N$ is the state change vector for each reaction R_j : v_{ij} represents the stoichiometry, 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 propensities, derived from the law of mass-action: the control input in particular directly affects the affinity term a_1 .

In order to synthesize a control architecture, we have adopted a discrete-time simulation framework. This has required a Δt -time sampling of both the RRE and the CME. The latter is achieved by a simple adaptation of the Stochastic Simulation Algorithm (SSA) [6]. Such a time discretization leads to the characterization of the model dynamics via one-step probability distributions, to be later used in the context of the model in (4). [meaning of last sentence?]

In order to prevent cell adaptation, our controller needs to comply with particular timing constraints: the valve should remain ON at most 8 minutes [actually between 5 and 8 minutes] and two stresses must be separated by at least 20 minutes (see Appendix A.1 for more details) [15].

Fluorescence measurements taken on every cell upon execution of the full image processing chain are modelled as

$$y_k = YFP_k + e_k, \quad e_k = (e_a + e_b \cdot YFP_k)\eta_k, \quad (3)$$

where y_k is the measurement at time k for a given cell, and η_k , with $k = 1, \dots, T$, 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.

3 Single-cell control with full state information

We consider the control of gene expression as a model-based optimal control problem, assuming in this section the availability of full-state information, that is, of the values of all the variables. Above we have formulated two models: a stochastic discrete-state one and a deterministic continuous-state one. For both cases, we will propose a control synthesis architecture based on the classical dynamic programming (DP) paradigm. As deterministic controllers have been previously used for the control of single cell dynamics [15], we use this approach as a benchmark against which the performance of the stochastic controller is later compared.

The goal of the control synthesis problem is to track a given profile of the protein concentration over a finite time horizon T . As the classical DP suffers from the curse of dimensionality, we employ an approximate DP method called Fitted Q-Iteration (FQI) [4], tailored here to the finite-horizon setting. In short, the FQI algorithm applies the idea of the fitted 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 to cover the entire state space.

Optimal controller synthesis via DP Let us recall that we work with models in discrete time, here indexed by k . 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:

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 1, \dots, T-1, \quad (4)$$

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: the recursive dynamics in (4) can be equivalently expressed by a conditional distribution $x_{k+1} \sim P(\cdot | x_k, u_k)$ that can be derived from a discrete-time version of the CME we discussed in the previous section, and where we explicitly emphasise the role of the chosen input.

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

$$Q_0^\pi(x_0, u_0) := \mathbb{E} \left[c_T(x_T) + \sum_{k=1}^{T-1} c_k(x_k, \pi_k(x_k)) \right], \quad (5)$$

where c_T is the terminal cost, a function only of the state variable. We are interested in the policy π^* that minimizes the cost as:

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

This can be obtained via DP by the backward recursion

$$Q_k^*(x, u) = \mathcal{T}Q_{k+1}^*(x, u), \quad (6)$$

initialised at c_T , and where \mathcal{T} is an operator acting on functions $H : X \times U \rightarrow \mathbb{R}$ as follows:

$$\mathcal{T}H(x, u) := c(x, u) + \inf_{u' \in U} \mathbb{E}H(f(x, u, w), u'). \quad (7)$$

The optimal policy is computed as

$$\pi_k^*(x) \in \arg \min_{u \in U} Q_{k+1}^*(x, u), \quad k = 0, \dots, T-1. \quad (8)$$

The Q -iteration in (6)-(8) 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.

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

$$\mathcal{F} = (x^i, u^i, c^{ij}, y^{ij}), \quad i = \{1, \dots, m_x\}, j = \{1, \dots, m_y\}, \quad (9)$$

where $x^i \in X$ is the instance of the current state, $u^i \in U(x^i)$ is the corresponding action, $y^{ij} \in X$ 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 y^{ij} , m_x is the number of current states, m_y is the number of successor states that are needed for the evaluation of the expectation operator in (7) using Monte-Carlo integration.

We adopt an offline approach owing to the computational complexity of the optimisation problem, and the stringent online time requirements. Using the batch of samples in (9), 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 $\hat{T}_{\mathcal{F}}$ as defined in (10): namely the value of $\mathcal{T}\hat{Q}_{k+1}$ is estimated as $\hat{T}_{\mathcal{F}}\hat{Q}_{k+1}$, for all x^i , $i = 1, \dots, m_x$.
- The second step involves fitting the approximation function \hat{Q}_k to $\hat{T}_{\mathcal{F}}\hat{Q}_{k+1}$: the optimal fit \hat{Q}_k is achieved by means of a regression algorithm.

The behaviour, generalisation capability, and computational complexity of the FQI method is heavily dependent 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 were tuned manually through trial and error. These parameters are crucial to determine the trade-off between the training error minimization, the smoothness and the generalization. We assume that the regression algorithm is fixed and denoted by \mathcal{G} the corresponding space of test functions $G : X \times U \rightarrow \mathbb{R}$. For a given tuple \mathcal{F} we denote

$$\hat{T}_{\mathcal{F}}H(x^i, u^i) := \inf_{u' \in U(x^i)} \frac{1}{m_y} \sum_{j=1}^{m_y} [c^{ij} + H(y^{ij}, u')] \quad (10)$$

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

FQI for the deterministic model An abstract discrete-time deterministic model is a special case of (4) where the update law f does not depend on the noise variable w . In our work, we refer to the deterministic dynamics 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 $m_y = 1$, so that only one successor state is needed for each instance of the current state. As a result,

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

One can therefore directly apply Algorithm 1 in the deterministic case.

Practical implementation of the stochastic FQI via receding horizon

Although the FQI for the deterministic model works well within our setup, the FQI algorithm for the stochastic model over the entire experimental duration T has been found to be 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, which are absent in the deterministic case. To overcome this issue, we embedded the FQI algorithm into a receding horizon strategy, resulting in a stochastic MPC-like scheme (see Algorithm 2 in Appendix) [1]. In short, over a finite prediction horizon $T_p \ll T$, the Q -functions are approximated offline using Algorithm 1. [Check correctness] 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 measurements are assumed to follow model (3). Hence, 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. However, the strategy can be generalized and 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 this 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, Unscented Kalman Filtering (UKF) [16], based on a continuous-valued approximation of the CME model known as the Chemical Langevin Equation (CLE) [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 circumscribe 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 $x^d = (pSTL1^{off}, pSTL1^{on}, CR \cdot pSTL1^{on})$ and $x^c = (mRNA, YFP)$. Consider a model where the dynamics of x^d (not depending on x^c) are left unchanged, while for any given trajectory of x^d , the dynamics of x^c are approximated by the Langevin equation

$$dx_i^c = \sum_{j=1}^M v_{ij}^c a_j(x^c, x^d) dt + \sum_{j=1}^M v_{ij}^c \sqrt{a_j(x^c, x^d)} dW_j, \quad i = 1, 2. \quad (11)$$

The i bottom index was previously used to denote time as k – confusing? Here, for $j = 1, \dots, M$, the W_j are independent Wiener processes and v_j^c is the sub-vector 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 and combining importance (particle) filtering with UKF is obtained by the following rationale. At time t_{k-1} , let $\hat{x}_{k-1|k-1}^c$ be the estimate of the current state x^c based on measurements y_0, \dots, y_{k-1} , and let $\hat{x}_{k-1|k-1}^{d,i}$,

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 $\hat{x}_{k-1}^{d,i}(t)$, with $t \in [t_{k-1}, t_k)$, is generated by stochastic simulation of the discrete-state dynamics starting from $\hat{x}_{k-1|k-1}^{d,i}$. Over the same time horizon, for every i , mRNA and protein state predictions $\hat{x}_{k-1}^{c,i}(t)$ are computed along trajectory $\hat{x}_{k-1}^{d,i}(t)$ via UKF. When the next protein measurement y_k becomes available, based on measurement model (3), an importance weight w_i , proportional to the likelihood of y_k given the hypothetical state value $\hat{x}_{k-1}^{c,i}(t_k)$, 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 $\hat{x}_{k-1}^{c,i}(t_k)$ are updated to estimates $\hat{x}_{k|k}^{c,i}$ 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 $\hat{x}_{k|k}^c$ as well as an ensemble (Maximum-A-Posteriori) estimate $\hat{x}_{k|k}^d$ for the discrete state are computed as

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

where $1_z(\cdot)$ is the indicator function. For control purposes, these are the estimates that are passed to the controller (provided rounding of the entries of $\hat{x}_{k|k}^c$ to the nearest integers). To proceed for the next iteration of the algorithm, the new putative values of the discrete state $\hat{x}_{k|k}^{d,j}$, with $j = 1, \dots, N$, are set equal to the result of N independent random extractions from the pool of particles $\{\hat{x}_{k-1}^{d,i}(t_k)\}_{i=1, \dots, N}$, 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 experimental time 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 small number of particles N much smaller than traditional particle filter implementations is expected to suffice. Empirical evaluation (not reported here) has led us 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 $\hat{x}_{k-1}^{d,i}$. Contrary to the control module, resorting to time discretization is not needed, although it can be considered towards higher computational efficiency.

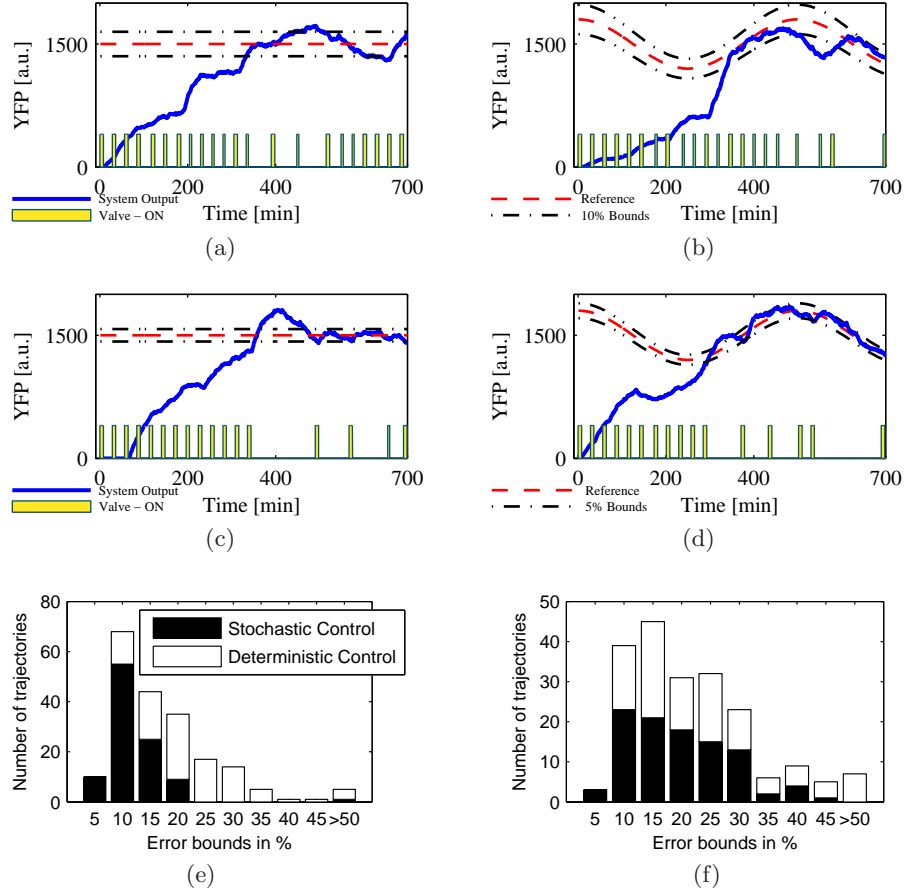


Fig. 2: Comparison of stochastic and deterministic control schemes in the full information case, run over the probabilistic model. (a)-(b) Deterministic controller tracking the desired profiles with a shown deviation of 10% from reference trajectories. (c)-(d) Stochastic controller showing improved performance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

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. 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 was successfully able to track the signals (see Appendix A.3 for details). This controller was then used over the stochastic CME model. At the maximum, the controller was able to track the reference signal to within a deviation of 10% as shown in Fig. 2(a)(b). The deterministic controller was then replaced with the stochastic controller (see Appendix A.2 for implementation details) and it was found that the stochastic controller was able to track the reference signal to within a deviation of 5% from the reference trajectory (Fig. 2(c)(d)). Owing to the stochastic nature of the system, the stochastic controller was not able to keep all the trajectories to within a deviation of 5% from the reference trajectory. To get a comprehensive comparison of the performance of the stochastic controller over the deterministic controller, 100 runs of each algorithm were performed using Monte-Carlo simulations. To measure the quality of the control, we use $\epsilon := \frac{1}{T-T_0} \sum_{k=T_0}^T |YFP_k - YFP_{ref,k}|$, 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(e)(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 the stochastic behavior of gene expression reactions.

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 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 Algorithm 3 has been used to extract information about the states of the gene expression network. The results of the state estimation with 50 particles is presented in Fig. 3(a). The filter finds it difficult to track the switching of the discrete states but is able to track the *mRNA* and *YFP* protein concentrations fairly accurately. We have then used the filter 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.

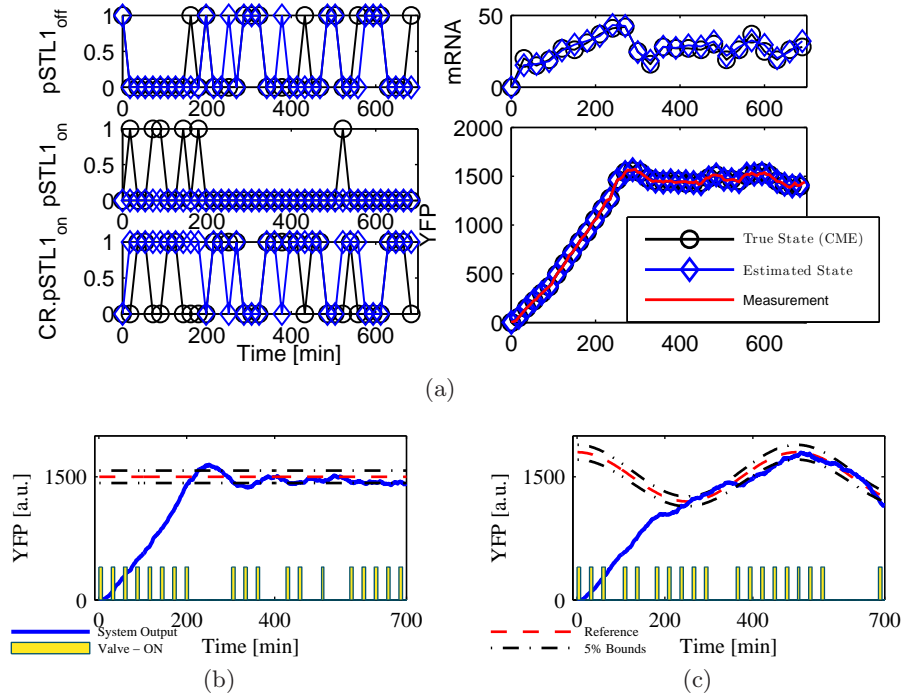


Fig. 3: Results of the stochastic control scheme run with the hybrid filter in the partial information case. (a) State estimation shows accurate results for mRNA and YFP, whereas the filter faces difficulties estimating the switching action of the promoter. (b)(c) Controller robustness over state estimation 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 been extensively studied recently, the control of gene expression using stochastic models has been barely addressed so far. This goal requires the non-trivial development of stochastic state estimators and controllers. We have demonstrated *in silico* that stochastic control has the potential to deliver superior performances in comparison to its deterministic counterpart. 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 [14, 17].

Acknowledgments. This research has been partly supported by a van Gogh grant from the Frans-Nederlandse Academie van Wetenschappen, 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. Hohmann. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiology and Molecular Biology Reviews*, 66(2):300–372, 2002.

10. 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.
11. A. Miliadis-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.
12. 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.
13. 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.
14. 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.
15. J. Uhlenhof, 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.
16. 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.
17. 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.
18. C. Zechner, J. Ruess, P. Krenn, S. Pelet, M. Peter, J. Lygeros, and H. Koepl. 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 the Gene Expression pathway

We have adopted a discrete-time framework to model the system dynamics. The switching of the valve leading to the transient osmotic stresses is modelled using a discrete state variable $q \in \{0, 1\}$, where 0 and 1 indicate the valve in the OFF and ON positions respectively. The time constraints on the switching of the valve to prevent cell adaptation are modelled using continuous state variables $\theta \in [0, 20]$ and $\eta \in [0, 8]$, both of which act as counters: the associated dynamics are characterized by the following difference equations:

$$\begin{aligned}\theta_{k+1} &= \theta_k \cdot (1 - \Delta q) + [q_k = 0, \theta_k < 20] \cdot \Delta t \\ \eta_{k+1} &= \eta_k \cdot (1 + \Delta q) + [q_k = 1] \cdot \Delta t,\end{aligned}\tag{13}$$

where Δt is the discretization step and $[\cdot]$ is the Iverson bracket, defined as

$$[x \in A] = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

There is a known lag between the valve actuation and the actual change of the osmolarity of the cellular environment in the imaging chamber: this has been characterized in [15] and is shown in Figure 4. We use a variable h to model this lag via the following difference equation:

$$h_{k+1} = h_k + [2 \leq \eta_k \leq 3] \cdot \Delta t - \frac{1}{4}[2 \leq \theta_k \leq 6] \cdot \Delta t.\tag{14}$$

It was noted in [18] that the signal transduction exhibits neither significant stochastic fluctuations nor cell-to-cell variability, hence the Hog1 activation dynamics s_k has been assumed to be purely deterministic:

$$s_{k+1} = s_k(1 - \Gamma \cdot \Delta t) + \kappa h_k \Delta t.\tag{15}$$

The promoter pSTL1 is activated at a rate u_k , which is assumed to follow Hill-type kinetics as a function of the nuclear Hog1 enrichment s_k [18]:

$$u_k = \frac{(s_k + s_0)^{n_H}}{K_d^{n_H} + (s_k + s_0)^{n_H}}.\tag{16}$$

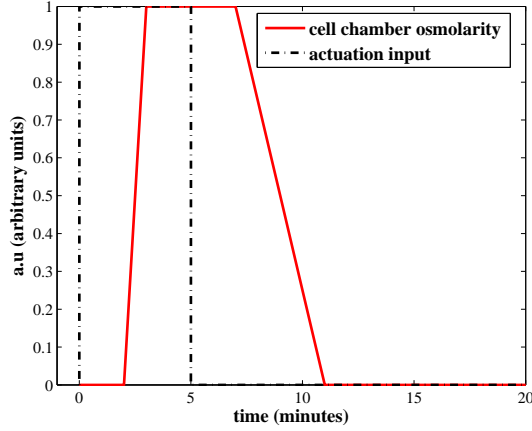


Fig. 4: Temporal evolution of the position of the microfluidic valve (0/1: normal/hyper-osmotic medium) and the osmolarity of the cellular environment.

A.2 Implementation details of the FQI algorithm over the stochastic CME model

For the stochastic rolling horizon control approach, the samples x^i were drawn corresponding to a single system trajectory. The trajectory was generated by simulating the system using the discrete time version of the SSA detailed in the section ?? above. For each x^i , 250 tuples ($f_s = 250$) of the form (x^i, u^i) were generated. For each tuple, the system was simulated 100 times ($m_s = 100$) to obtain the next state y^{ij} to evaluate the Monte-Carlo integration. The cost c^{ij} was computed according to (17) and a single batch of 25,000 tuples ($\mathcal{F}_s = 25,000$) was obtained. The optimization was performed for a prediction horizon N_p of 8 minutes and a time horizon N_t of 700 minutes. The squared bandwidth σ^2 and the regularization parameter γ of the regression algorithm were tuned by a trial and error method as in the deterministic case and the parameters are listed below.

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

Table 1: Tuned LS-SVM parameters to track time varying and time constant profiles using the controller trained on the stochastic CME model.

The stochastic FQI algorithm and the stochastic MPC algorithm are detailed below

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

- 1: Initialize the parameters of the regression algorithm σ^2 and γ and set \hat{Q}_T to 0
- 2: **for** $k := T - 1$ **to** 0 **do**
- 3: Estimate $\hat{T}_{\mathcal{F}}\hat{Q}_{k+1}$.
- 4: Find the fit that minimizes the 2-norm loss by means of a regression algorithm

$$\hat{Q}_k(x, u) = \arg \min_{G \in \mathcal{G}} \|G - \hat{T}_{\mathcal{F}}\hat{Q}_{k+1}\|_{\mathcal{F}}.$$

- 5: **end for**
-

Algorithm 2 Stochastic Model Predictive Control (T, T_p, \mathcal{F})

- 1: **for** $k:=1$ **to** T **do**
- 2: Initialize the parameters of the regression algorithm σ^2 and γ and set Q_T to 0
- 3: **for** $l := k + T_p$ **to** k **do**
- 4: Estimate $\hat{T}_{\mathcal{F}}\hat{Q}_{l+1}$.
- 5: Find the fit minimizing the 2-norm loss by means of a regression algorithm

$$\hat{Q}_l(x, u) = \arg \min_{G \in \mathcal{G}} \|G - \hat{T}_{\mathcal{F}}\hat{Q}_{l+1}\|_{\mathcal{F}}.$$

- 6: **end for**
 - 7: **end for**
-

A.3 Implementation details of the FQI algorithm over the deterministic RRE model

For the deterministic control approach presented in Section 3, 400 tuples were generated corresponding to a single trajectory. The trajectory was obtained by simulating reactions of the gene expression network using the RRE. A time horizon of 700 minutes was considered and the regression algorithm was implemented using the LS-SVM MATLAB toolbox in [13]. The *fixed size* LS-SVM model, provides two parameters for tuning, the squared bandwidth σ^2 and the regularization parameter γ . The parameters were tuned manually using a trial and error method and are listed in Tables 2 and 3 below.

A stage-dependent cost function has been chosen as below:

$$c_k(YFP_k) = |YFP_k - YFP_{ref,k}|. \quad (17)$$

The function penalises deviations from a reference profile YFP_{ref} , with a minimum that is achieved when the protein concentration of the system equals the reference protein concentration. For the deterministic control approach, the deterministic version of the FQI algorithm was 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 Bandwidth (σ^2)	Regularization Parameter (γ)
700 - 651	40000	$1e - 1$
650 - 601	40000	$1e - 2$
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 2: Tuned LS-SVM parameters to track a set-point of 1500 (a.u.) using the controller trained on the deterministic RRE model.

Time Horizon (T)	Squared Bandwidth (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 3: Tuned LS-SVM parameters to track the sinusoidal reference signal using the controller trained on the deterministic RRE model.

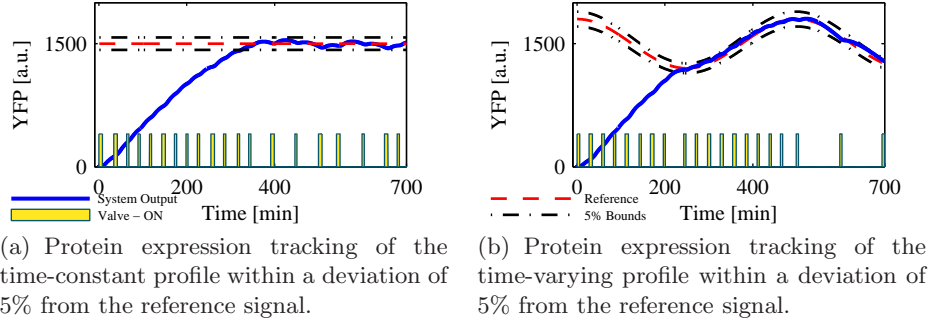


Fig. 5: Results for the deterministic control scheme in the full information case.

A.4 Parameters employed in the simulation and analysis of the model

Parameter	Value	Parameter	Value
θ_0	0 (a.u.)	Δt	0.008 (<i>min</i>)
η_0	0 (a.u.)	T_s	1 (<i>min</i>)
h_0	0 (a.u.)	Γ	0.9225 (a.u.)
s_0	0 (a.u.)	κ	0.3968 (a.u.)
$(pSTL1_{off})_0$	1 (a.u.)	K_d	0.34906 (a.u.)
$(pSTL1_{on})_0$	0 (a.u.)	n_H	2.1199 (a.u.)
CR_0	102.51 (a.u.)	s_0	0.0027998 (a.u.)
$(CR \cdot pSTL1_{on})_0$	0 (a.u.)	$var(CR_0)$	1.5
$mRNA_0$	0 (a.u.)	$var(c_6)$	$9.7e - 4$
YFP_0	0 (a.u.)	c_5	$12.256 (min)^{-1}$
c_1	$23.604 (min)^{-1}$	c_6	$0.36113 (min)^{-1}$
c_2	$180.03 (min)^{-1}$	c_7	$0.025091 (min)^{-1}$
c_3	$0.024559 (min)^{-1}$	c_8	$0.003354 (min)^{-1}$
c_4	$0.9384 (min)^{-1}$	$cov(CR_0 c_6)$	$1.2e - 6$
e_a	$1.0115 (min)^{-1}$	e_b	$0.0037 (min)^{-1}$

Table 4: Initial concentrations and rate parameters of the model

A.5 Hybrid Estimation Algorithm

Algorithm 3 Hybrid Filter for Estimation of the Model States

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