An automated model reduction tool to guide the design and analysis of synthetic biological circuits

Ayush Pandey Richard M. Murray

***Abstract*—We present an automated model reduction algorithm that uses quasi-steady state approximation based reduction to minimize the error between the desired outputs. Additionally, the algorithm minimizes the sensitivity of the error with respect to parameters to ensure robust performance of the reduced model in the presence of parametric uncertainties. We develop the theory for this model reduction algorithm and present the implementation of the algorithm that can be used to perform model reduction of given SBML models. To demonstrate the utility of this algorithm, we consider the design of a synthetic biological circuit to control the population density and composition of a consortium consisting of two different cell strains. We show how the model reduction algorithm can be used to guide the design and analysis of this circuit.**

I. Introduction

Model reduction is an indispensable tool in most engineering applications where mathematical models are used for design and analysis of systems. Due to the high complexity of physical systems, it is possible to create arbitrarily large and detailed mathematical models of the different processes, however, to use these models for design of systems can be challenging and the analysis of the system response using these models can be convoluted in various ways. Hence, it is common to model various processes that we are interested in using a "reduced" order model. A reduced order model is a lower dimensional model that has a simple representation, is computationally much faster, and is easier to use for system design compared to any other higher order, more complex mathematical model. For biological systems, usually, we do not have enough information about the various mechanistic interactions to model these processes effectively. This limitation is manifested in the lack of information about the parameters of a biological model. Due to these reasons, it is even more important to be able to have reduced models to model various biological processes. There are multiple ways and scales at which biological systems are modeled such as using the mass-action kinetics of the chemical reactions [1], using the chemical master equation [2], the chemical Langevin equation [3], or using methods from statistical mechanics [4]. Mass-action kinetic based modeling uses reaction rate equations where dynamics of various processes are approximated by modeling the concentration of various species. With mass-action kinetics based modeling of chemical reactions describing a biological system, it is possible to create models that are low-dimensional and simple compared to other methods. The most widely used model reduction methods utilize the separation of time scales [5] in a model, or use balanced truncation [6–8] that transforms the original model into a lower dimensional system while preserving the input-output behavior. A combined approximate method using singular perturbation theory and balanced truncation is developed in [9]. Other methods such as parameter sensitivity analysis based reduction [10, 11], optimization based methods [12, 13], lumping of parameters [14] of the model, non-dimensionalization [15], and conservation analysis are also common. In [16] some of these different methods have been combined together for an improved model reduction method. Generally, the choice of model reduction is dependent on the kind of application and is driven by the questions we want to answer using a reduced model. See [17–19] for a detailed review of some of the well known model reduction methods with applications to biological systems and the current best practices.

Among the methods based on time-scale separation, the Computational Singular Perturbation (CSP) [5] algorithm is one of the most popular numerical methods that identifies the fast and slow time-scales for a model. But, apart from being stiff and laborious to solve, it is not always clear for many applications how fast and slow time-scales can be separated. The Intrinsic Low-Dimensional Manifold (ILDM) [20, 21] is another numerical method based on time-scale separation in which the original model is linearized and then the Jacobian of the resulting linearized system is transformed to identify slow and fast states. But, state transformations are introduced in this method due to which the structure and the physical meaning in the original model are lost. Although balanced truncation based methods can give good error bounds, we face a similar problem since the transformations introduced might lead to the loss of the structure of the original model. Hence, it would be difficult to use such methods to guide the design of biological systems where experimental connections of the model states and parameters are important.

Quasi-steady state approximation (QSSA) [22–25] is one of the most widely used approaches used in practice. The theory driving QSSA based reduction is derived from singular perturbation theory [26] of nonlinear dynamical systems. We will discuss singular perturbation theory in more detail later as our algorithm in this paper is motivated from this method of model reduction. However with QSSA, one needs to be

A.P. is with the Department of Electrical Engineering, California Institute of Technology. R.M.M. is with the Department of Control and Dynamical Systems and Department of Bioengineering, California Institute of Technology, Pasadena, CA

careful when collapsing states into algebraic relations, as it might not always give correct results. A counterexample in [1] elucidates this in detail. The rigorously correct way of performing a QSSA would be when it can be formulated in singular perturbation theory framework. But, it is often the case that the conditions with singular perturbation are too strict and it is not possible for many applications to formulate the model in this framework. Hence, it is important to carefully navigate this gap between the rigorously correct singular perturbation and using QSSA directly. Our algorithm in this paper works within this gap and finds reduced order models that minimize the error in desired outputs along with ensuring robust performance of the reduced models.

In this paper, we present a structured automated model reduction algorithm that systematically finds a reduced model while retaining the structure of the original model. We start by briefly discussing some of the preliminary results and formulating the problem in Section II before presenting our main results in Section III. We first present the results for systems with linear dynamics and then move on towards the extension of the results for nonlinear dynamics. Subsequently, we give an algorithm that builds on the model reduction procedure for nonlinear dynamics developed. In Section IV, we consider two examples to demonstrate the utility of the developed algorithm. First, we consider a simple example of a toggle switch circuit and then present the model reduction of a population density and composition control circuit where we show how our algorithm plays an important role in identifying a reduced model that could be used for design and analysis of this circuit.

II. Problem Formulation

Definition 1 (*Full model*)**.** For a state vector $x \in \mathbb{R}^n$, a vector consisting of all model parameters $\Theta \in \mathbb{R}^p$ and the vector of output measurements $y \in \mathbb{R}^k$, we define the nonlinear dynamic model

$$\begin{aligned} \dot{x} &= f(x, \Theta) \\ y &= Cx \end{aligned}$$

for initial conditions $x(0) = x_0 \in \mathbb{R}^n$.

Definition 2 (*Reduced model*)**.** For a full model as defined above, we define a reduced model with a lower dimensional state vector $\hat{x} \in \mathbb{R}^{\hat{n}}$ where $\hat{n} < n$ as

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x}, \Theta) \\ \hat{y} &= \hat{C}\hat{x} \end{aligned}$$

for initial conditions $\hat{x}(0) = \hat{x}_0 \in \mathbb{R}^{\hat{n}}$ and $\hat{y} \in \mathbb{R}^k$.

Note that in the reduced model we have reduced the number of states (from n to $\hat{n}$) but we have the same number of outputs that depend only on the reduced states. It is clear that in the reduced model some parameters from the full model will be lumped together, hence reducing the number of parameters in the model as well. We will see later how this structure of the reduced model is beneficial for synthetic biological circuit models. Further, we will need the following definitions to formulate the model reduction problem that we solve in this paper.

Definition 3 (*Sensitivity coefficients*)**.** For any state, x_i for all $i = 1, 2, ..., n$, the sensitivity coefficient for the state with respect to a parameter $\theta_j \in \Theta$ for all $j = 1, 2, ..., p$ is defined as

$$s_{ij} = \frac{\partial x}{\partial \theta_j}.$$

We also define $S_j = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_j} & \frac{\partial x_2}{\partial \theta_j} & . & . & . & \frac{\partial x_n}{\partial \theta_j} \end{bmatrix}^T$, the vector of sensitivity coefficients for all states with respect to the parameter θ_j

Lemma 1. *The sensitivity coefficients satisfy a linear differential equation given by,*

$$\dot{S}_j = JS_j + Z_j,\ S(0) = S_0,$$

where $Z_j \in \mathbb{R}^n$ consists of partial derivatives $\frac{\partial f_i}{\partial \theta_j}$ and $J \in \mathbb{R}^{n \times n}$ is the Jacobian matrix,

$$Z_j = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_j} \\ \frac{\partial f_2}{\partial \theta_j} \\ . \\ . \\ . \\ \frac{\partial f_n}{\partial \theta_j} \end{bmatrix}, \quad J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & . & . & . & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & . & . & . & \frac{\partial f_2}{\partial x_n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & . & . & . & \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

Proof. Follows from chain rule of differentiation [27]. □

Similarly, for the reduced model we can define $\hat{S}_j$, $\hat{Z}_j$ and $\hat{J}$ and we have,

$$\dot{\hat{S}}_j = \hat{J}\hat{S}_j + \hat{Z}_j.$$

Remark. The initial conditions for the sensitivity coefficients

$$S_0 = \frac{\partial x_0}{\partial \theta}$$

will be equal to zero if the initial conditions x_0 are independent of parameters. If the initial conditions x_0 are dependent on parameters then the corresponding initial condition for the sensitivity coefficients will be equal to 1.

As discussed in the previous section, in our approach we carry over the essential information from the full model to the reduced model by collapsing some of the states' dynamics into algebraic relations. This process is reminiscent of singular perturbation based model reduction algorithms. In the next theorem, we briefly introduce this topic and refer the interested reader to [26] for more theoretical details and [5] for its applications to model reduction.

Theorem 1. *Suppose we have*

$$\begin{aligned} \dot{x}_1 &= f(x_1, x_2, \epsilon), & x_1(0) &= x_{10}, & x_1 &\in \mathbb{R}^{n_1}, \\ \epsilon\dot{x}_2 &= g(x_1, x_2, \epsilon), & x_2(0) &= x_{20}, & x_2 &\in \mathbb{R}^{n_2}, \end{aligned}$$

for $0 < \epsilon \ll 1$ and both functions $f(x_1, x_2, 0), g(x_1, x_2, 0)$ are well defined. With these conditions, if we are only interested in the slow time-scale dynamics (i.e. x_1), we can write the reduced model for this system described by the set of differential equations above. Then,

$$\dot{\hat{x}} = f(\hat{x}, h(\hat{x}), 0) \qquad \hat{x}(0) = x_{10}, \quad \hat{x} \in \mathbb{R}^{n_1},$$

describe the fast dynamics of the system where $x_2 = h(x_1)$ is called the slow manifold. If the eigenvalues of the Jacobian obtained by linearizing $g(x_1, x_2)$ with respect to x_2 on the slow manifold have negative real part, then there exist positive constants ϵ^* and M such that

$$\|x_1 - \hat{x}\| \leq M\epsilon, \ \|x_2 - h(\hat{x})\| \leq M\epsilon.$$

provided $\epsilon < \epsilon^*$. Hence, the reduced model converges locally to the full model in the limit.

Proof. See [26]. □

In the model reduction algorithm that we present in the next section, we use the following definition for the error in reduction and the sensitivity of the error.

Definition 4. We define the error in reduction for the output y_i as $e_i = y_i - \hat{y}_i$ for all $i = 1, 2, ..., k$. We also define the sensitivity of this error with respect to the model parameter $\theta_j \in \Theta$ as $s_{e_{ij}} = \frac{\partial e_i}{\partial \theta_j}$ for all $j = 1, 2, ..., p$.

We formulate the desired properties of the reduced models with a view towards how we would like to use these reduced models in design and analysis of synthetic biological circuits. There are two major ways in which reduced models are useful. Firstly, we would like to have simple models that represent the design of a circuit reasonably well so as to use these models to get insights about and improve the design of the various components of the circuit. Secondly, since it is often the case for biological circuits that we can only observe a few states (for e.g. fluorescent outputs), hence, models are often non-identifiable. Reduced models with lower number of states and lumped parameters would decrease the number of non-identifiable parameters in a model. With these applications in mind, we will formulate our assumptions accordingly for the model reduction algorithm that we develop.

Given the dynamics of a full model, our objective is to minimize the weighted norm of the error e (as in Definition 4) for all time. But, if for a given model we have multiple reduced models that satisfy the desired error tolerance, we would like to have a reduced model that is robust to the uncertainties in the model parameters. We would also like the reduced model to have the least number of states and parameters, to the extent possible. The model reduction algorithm presented in this paper takes into account all of these objectives.

III. Results

A. Linear dynamics

Consider the case when the dynamics of the full model are linear of the following form,

$$\dot{x} = A(\Theta)x, \qquad x(0) = x_0,$$

and we want to construct,

$$\dot{\hat{x}} = \hat{A}(\hat{\Theta})\hat{x}, \qquad \hat{x}(0) = \hat{x}_0,$$

where we assume that A and $\hat{A}$ are stable ($A \in \mathbb{R}^{n\times n}$ and $\hat{A} \in \mathbb{R}^{\hat{n}\times\hat{n}}$) and dependent on the parameters. For outputs, we have $y = Cx$ and $\hat{y} = \hat{C}\hat{x}$. To construct the reduced models, we can choose the states that we would like to collapse and replace their dynamic equation by an algebraic relationship and create a vector of all such collapsed state variables x_c. This procedure to obtain model reduction is similar to the singular perturbation based model reduction in Theorem 1. We can write

$$x = T \begin{bmatrix} \hat{x} \\ x_c \end{bmatrix} = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} \hat{x} \\ x_c \end{bmatrix},$$

where T is a permutation matrix consisting of only zeros and ones with the condition that there can only be one non-zero element in a row and a column. Note that here $T_1 \in \mathbb{R}^{n\times\hat{n}}$ and $T_2 \in \mathbb{R}^{n\times(n-\hat{n})}$. With this structure for T, we ensure that the states do not lose their meaning in the reduced model and reduced states are a strict subset of the states of the full model. The model reduction problem then is that of choosing the matrix T so that the error in reduction is minimized and the reduced model is robust to parametric uncertainties. Towards that end, we will first consider only the minimization of the two norm of the error for all time. We will use the results from [28] for this case. Subsequently, we will extend these results to improve the model reduction algorithm so that the performance of the reduced models is robust to uncertainties in parameters.

Define $\bar{x} = \begin{bmatrix} x & \hat{x} \end{bmatrix}^T$, then we can write e from Definition 4 as $e = \begin{bmatrix} C & -\hat{C} \end{bmatrix} \bar{x} \triangleq \bar{C}\bar{x}$. We can write the dynamics of $\bar{x}$,

$$\dot{\bar{x}} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix} \bar{x} := \bar{A}\bar{x}.$$

Our objective is to minimize $\|e\|_2^2 = \int_0^\infty e^T(t)e(t)dt$ from an initial condition $\bar{x}_0 = \begin{bmatrix} x_0 \\ \hat{x}_0 \end{bmatrix}$.

Theorem 2. *Suppose that there is a matrix $P = P^T \succeq 0$ that solves the continuous-time Lyapunov equation for the augmented system, $\bar{A}^T P + P\bar{A} + \bar{C}^T\bar{C} = 0$, then the norm of the error in reduction can be upper bounded by the largest eigenvalue of a matrix Q,*

$$\max_{\|\bar{x}_0\|_2^2 = 1} \|e\|_2^2 = \max(\lambda_Q),$$

where

$$Q = \begin{bmatrix} T_1^T P_{11} T_1 + T_1^T P_{12} + P_{21} T_1 + P_{22} & T_1 P_{11} T_2 + P_{21} T_2 \\ T_2^T P_{11} T_1 + T_2^T P_{12} & T_2^T P_{11} T_2 \end{bmatrix},$$

where we partition P as,

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},$$

so that $P_{11} \in \mathbb{R}^{n\times n}$ and $P_{22} \in \mathbb{R}^{\hat{n}\times\hat{n}}$.

Proof. See [28]. □

The above theorem gives us an upper bound on the error in model reduction and can be formulated as an optimization problem to solve for the "best" reduced model out of all possible permutation matrices T. With the next theorem, we will extend this result to improve the model reduction algorithm so that we are able to choose the reduced models that are least sensitive to uncertainties in parameters. We begin with a lemma that gives us the dynamics of the sensitivity of the error, which we then use in the theorem that follows that gives the main result for minimization of the sensitivity of the error for the linear case.

Lemma 2. *The sensitivity of error (Definition 4) can be described as an output of the linear dynamical system of the augmented sensitivity vector.*

$$\dot{\bar{S}} = \bar{A}\bar{S} + \left(\frac{\partial \bar{A}}{\partial\theta}\bar{x}\right),$$

where $\bar{S} = \begin{bmatrix} S \\ \hat{S} \end{bmatrix}$, then the sensitivity of the error is given by,

$$S_e = \bar{C}\bar{S}.$$

Proof. Using the results from lemma 1 for $\bar{S}$,

$$\dot{S} = \frac{\partial Ax}{\partial\theta}$$

$$\dot{S} = AS + \frac{\partial A}{\partial\theta}x.$$

Similarly,

$$\dot{\hat{S}} = \hat{A}\hat{S} + \frac{\partial \hat{A}}{\partial\theta}\hat{x},$$

which gives $\dot{\bar{S}} = \bar{A}\bar{S} + \left(\frac{\partial \bar{A}}{\partial\theta}\bar{x}\right)$ and

$$\dot{S}_e = \frac{\partial}{\partial\theta}(CAx)) - \frac{\partial}{\partial\theta}\left(\hat{C}\hat{A}\hat{x}\right).$$

Simplifying, we get the desired result, $S_e = \bar{C}\bar{S}$ and $\dot{S}_e = \bar{C}\dot{\bar{S}}$. □

For the next step of our algorithm, our goal is to minimize $\|S_e\|_2^2 = \int_0^\infty S_e^T(t)S_e(t)dt$. From computational point of view, if we were to set out to implement this we would have to perform the sensitivity analysis of the full model and all possible reduced models for all time to get $\bar{S}$ and then use the previous lemma to associate the error sensitivity cost with each reduced model. Clearly, this procedure would be computationally very inefficient. The next theorem gives us a result to simplify this computation by giving a simpler way to get an upper bound on the norm of the sensitivity of the error.

We can use the results of this theorem to compute this bound for all possible reduced models and then make a conclusion about the best reduced model along with the results obtained for the minimization of the error.

Theorem 3. *Suppose that there is a matrix $P = P^T \succeq 0$ that solves the continuous time Lyapunov equation $\bar{A}^T P + P\bar{A} = -\bar{C}^T\bar{C}$, then the norm of the sensitivity of the error can be upper bounded as follows,*

$$\|S_e\|_2^2 \leq \lambda_{\max}(P) + 2N \max_t \left\| \begin{bmatrix} \hat{x} \\ x_c \end{bmatrix}^T \bar{M} \begin{bmatrix} \frac{\partial \hat{x}}{\partial\theta} \\ \frac{\partial x_c}{\partial\theta} \end{bmatrix} \right\|_2,$$

where N is a positive constant, and

$$\bar{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},\ P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},$$

$$M_{11} = T_1^T \frac{\partial A}{\partial\theta}^T P_{11}T_1 + T_1^T \frac{\partial A}{\partial\theta}^T P_{12} + \frac{\partial \hat{A}}{\partial\theta}^T P_{21}T_1 + \frac{\partial \hat{A}}{\partial\theta}^T P_{22},$$

$$M_{12} = T_1^T \frac{\partial A}{\partial\theta}^T P_{11}T_2 + \frac{\partial \hat{A}}{\partial\theta}^T P_{21}T_2,$$

$$M_{21} = T_2^T \frac{\partial A}{\partial\theta}^T P_{11}T_1 + T_2^T \frac{\partial A}{\partial\theta}^T P_{12},$$

$$M_{22} = T_2^T \frac{\partial A}{\partial\theta}^T P_{11}T_2,$$

where $M_{11} \in \mathbb{R}^{\hat{n}\times\hat{n}}$ and $M_{22} \in \mathbb{R}^{\hat{n}\times(n-\hat{n})}$.

Proof. We will prove this theorem in the more general case of nonlinear dynamics in the next section. The proof for the linear case will follow as a special case. □

B. Nonlinear dynamics

For error minimization in the case of nonlinear dynamics, we no longer have the simple analytical results as in Theorem 2. For networked systems with large n, i.e. for models with a large number of states the combinatorial procedure of forming possible reduced models and then calculating the error in reduction will be an inefficient algorithm with exponential time-complexity. A greedy algorithm method based on SOSTOOLS [29] to bound the error in reduction for such systems is given in [28]. It gives sub-optimal results and an approximate bound on the error but it is computationally efficient for such systems. However, for our application, we consider minimal ODE models that use Hill functions to approximate the reaction model dynamics in a way that is suitable for experimental design and analysis. With these models, we often do not have a large n. Moreover, usually, the states are directly related to the different components of the circuit that we are designing and hence it would not be wise to take a greedy algorithm approach with these models. So, for system dynamics where n is not large (< 20), we can computationally calculate the error in model reduction for all possible reduced order models in a brute force way. We discuss this in more detail later when we present the algorithm.

The second part of the model reduction method that we presented for the linear case is the minimization of the sensitivity of the error in model reduction. Using this, we can identify the reduced models that have robust performance in the presence of parametric uncertainties. For the nonlinear dynamics, we have similar results as obtained earlier for the linear case that we now present beginning with the next lemma that shows the sensitivity of error as the output of the augmented sensitivity system.

Lemma 3. *The sensitivity of error as defined in Definition 4 for the nonlinear dynamics can be described as an output of the linear dynamical system of the augmented sensitivity vector,*

$$\dot{\bar{S}} = \bar{J}\bar{S} + \bar{Z},$$

where $\bar{J} = \begin{bmatrix} J & 0 \\ 0 & \hat{J} \end{bmatrix}$ *and* $\bar{Z} = \begin{bmatrix} Z \\ \hat{Z} \end{bmatrix}$. *The error in sensitivity is then given by,*

$$S_e = \bar{C}\bar{S}.$$

We can also write the dynamics of the sensitivity of the error $\dot{S}_e = \bar{C}\bar{J}\bar{S} + \bar{C}\bar{Z}$.

Proof. The proof follows similar to the proof for Lemma 2 where now we have the Jacobians (J and $\hat{J}$) of the system dynamics and the matrices (Z and $\hat{Z}$) as defined in Section II instead of $A, \hat{A}$ and $\frac{\partial A}{\partial \theta}, \frac{\partial \hat{A}}{\partial \theta}$. □

It is possible to use this lemma to implement an algorithm that solves for the sensitivity of the error by integrating the error sensitivity dynamics equations along with the system dynamics. However, with the next theorem we show a result that gives a method to upper bound the norm of the sensitivity of the error that can be computationally more efficient than the brute force sensitivity analysis of all possible reduced order models.

Theorem 4. *Suppose that there is a matrix* $P = P^T \succeq 0$ *that solves the continuous-time Lyapunov equation for the system matrix* $\bar{J}$ *in the sensitivity system in Lemma 3, i.e.,* $\bar{J}^T P \bar{J} + P\bar{J} = -\bar{C}^T\bar{C}$, *then the norm of the sensitivity of the error for the nonlinear model reduction can be upper bounded as follows for some* $N > 0$,

$$\|S_e\|_2^2 \leq \lambda_{\max}(P) + 2N \max_t \left\| \begin{bmatrix} \frac{\partial f}{\partial \theta} \\ \frac{\partial \hat{f}}{\partial \theta} \end{bmatrix} Q_s \begin{bmatrix} \frac{\partial \hat{x}}{\partial \theta} \\ \frac{\partial x_c}{\partial \theta} \end{bmatrix} \right\|_2, \quad (1)$$

where we partition P,

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},$$

for $P_{11} \in \mathbb{R}^{\hat{n}\times\hat{n}}$, $P_{22} \in \mathbb{R}^{(n-\hat{n}\times(n-\hat{n})}$ *and*

$$Q_s = \begin{bmatrix} P_{11}T_1 + P_{12} & P_{11}T_2 \\ P_{21}T_1 + P_{22} & P_{21}T_2 \end{bmatrix}.$$

Proof. See Appendix A. □

C. Algorithm

In this section, we present a brute force algorithm based on the results presented in the previous section to compute the sensitivity of the error efficiently for all reduced models. Note that the set R in the algorithm is the set of all possible combinations of states that we attempt to reduce. This set is created by simply creating combinations of all states but also ensuring that the states corresponding to the outputs are never collapsed. For the purpose of implementation, the permutation matrix T in the algorithm consists of indices of states that will be collapsed (as opposed to the 0s and 1s as used above for the theoretical results, where 1 corresponded to the states that were collapsed). The time-complexity of the algorithm is

Algorithm 1: Automated model reduction algorithm

GetReducedModel (*SBML model, C, tol_e, tol_{S_e}*) **:**
- $\hat{x} \leftarrow [\,], x_c \leftarrow [\,], \hat{f} \leftarrow [\,], f_c \leftarrow [\,]$
- $R \leftarrow$ Set of all T's
- $f, x \leftarrow$ Convert SBML model to symbolic model
- **for** *j = 1:length(R)* **do**
  - T = R[j]
  - **for** *i = 1:length(x)* **do**
    - **if** *i in T* **then**
      - $x_c[i] \leftarrow x[i]$
      - $f_c[i] \leftarrow f[i]$
    - **else**
      - $\hat{x}[i] \leftarrow x[i]$
      - $\hat{f}[i] \leftarrow f[i]$
    - **end**
  - **end**
  - **for** *k = 1:length(x_c)* **do**
    - **for** *k = 1:length($\hat{f}$)* **do**
      - $\hat{f}[k] \leftarrow$ Solution of $f_c[i] = 0$ for $x_c[i]$
    - **end**
  - **end**
  - Solve ODEs for x and $\hat{x}$ and compute y and $\hat{y}$
  - $e[j] \leftarrow \|y - \hat{y}\|_2$
  - $S_e[j] \leftarrow \|S_e\|_2$ from equation (1)
- **end**
- **if** *min(e) < tol_e and min(S_e) < tol_{S_e}* **then**
  - index_min = index of min(e)
  - reduced_model = $\hat{f}$[index_min]
- **else**
  - reduced_model = None
- **end**
- **return** reduced_model

exponential with n, the size of the full model, since we search for the best reduced model over all possible combinations of the matrix T. However, it is possible to implement the algorithm efficiently when n is not large, that is for system dynamics that are not networked dynamics with a large number of states. Particularly, an estimate of the number of states of the full model for which it would be possible to use this algorithm would be $n < 20$. For our application of model

reduction of synthetic biological circuit models, this algorithm is suitable, since we often have coarse-grained models that use Hill function modeling [1] to obtain simple enough models that describe the system dynamics effectively. We will demonstrate the utility of the algorithm in the next section with a simple example first and then also for a circuit that implements population density and composition control of two cell strains in a consortium.

IV. Examples

We will start by considering a simple example of a toggle switch model.

A. Toggle Switch

1) Model: The toggle switch is a two species system where both proteins A and B repress each other. We consider the following toggle switch system where proteins TetR and LacI repress each other's expression. LacI binds with the promoter for protein TetR to prevent the transcription of the TetR gene to form its mRNA and hence repressing the production of protein TetR. Similarly, TetR binds with the promoter of the LacI gene to prevent its transcription to form mRNA, which in turn causes the repression in the expression of LacI protein. This simple two species transcriptional regulation system is shown in Fig. 1. It shows bistability — has two stable equilibrium

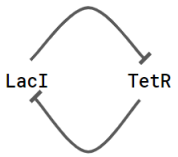


Fig. 1. A two species toggle switch where LacI and TetR repress each other's expression. We can control the strength of each repression using inducers, which effectively control the values of the parameters b_T and b_L in the model given in equation (2)

points and one unstable equilibrium [1]. Here we consider one of the stable equilibrium points and attempt to get a reduced model around this equilibrium. The derivation for the nonlinear ODEs of this system is given in [1]. The final model is

$$\begin{aligned}\dot{m}_T &= \frac{Kb_T^n}{b_T^n + p_L^n} - d_T m_T,\\ \dot{m}_L &= \frac{Kb_L^n}{b_L^n + p_T^n} - d_L m_L,\\ \dot{p}_T &= \beta_T m_T - \delta_T p_T,\\ \dot{p}_L &= \beta_L m_L - \delta_L p_L.\end{aligned} \tag{2}$$

If we consider the two proteins as output species then

$$C = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}.$$

According to the algorithm proposed in the previous section, the possible number of reduced models will be three. The corresponding list R will be $\{(1,3,4),(2,3,4),(3,4)\}$ where 1 corresponds to the state m_T and so on for the other three states. For a particular set of parameters where $\beta_i, \delta_i \ll b_i, d_i$, we can express the model in equation (2) in the singular perturbation theory framework defined earlier in Theorem 1. From singular perturbation theory based reduction, we get the following reduced model (See [1])

$$\hat{f} = \begin{bmatrix}\frac{\beta_T K b_T^n}{d_T(b_T^n + \hat{p}_L^n)} - \delta_T \hat{p}_T\\ \frac{\beta_L K b_L^n}{d_L(b_L^n + \hat{p}_T^n)} - \delta_L \hat{p}_L\end{bmatrix}.$$

We can use this as a proposed reduced model to validate with the algorithm and the theory we developed in this paper. Particularly, we will see how the conditions when this reduced model is valid imply that the sensitivity of the error in reduction converges to zero. The singular perturbation based reduction is a special case of the model reduction approach using the algorithm presented in the previous section. For this special case, we will see that the error sensitivity with respect to parameters takes a simple form that we can evaluate.

$$e_1 \triangleq p_T - \hat{p}_T,\ e_2 \triangleq p_L - \hat{p}_L,$$

$$f_{m_1} \triangleq \frac{Kb_T^n}{b_T^n + p_L^n},\ f_{m_2} \triangleq \frac{Kb_L^n}{b_L^n + p_T^n}.$$

For a parameter $\theta_j \in \Theta$ and $i = 1, 2$, we have,

$$S_j = \begin{bmatrix}\frac{\partial m_i}{\partial \theta_j}\\ \frac{\partial p_i}{\partial \theta_j}\end{bmatrix},\ \hat{S}_j = \begin{bmatrix}\frac{\partial \hat{p}_i}{\partial \theta_j}\end{bmatrix},$$

$$Z_j = \begin{bmatrix}\frac{\partial}{\partial \theta_j}\left(f_m(y) - d_i m_i\right)\\ \frac{\partial}{\partial \theta_j}\left(\beta_i m_i - \delta_i p_i\right)\end{bmatrix},\ \hat{Z}_j = \begin{bmatrix}\frac{\partial}{\partial \theta_j}\left(\beta_i f_m(\hat{y}) - \delta_i \hat{p}_i\right)\end{bmatrix},$$

$$J = \begin{bmatrix}-d_i & \frac{\partial f_m}{\partial p_i}\\ \beta_i & -\delta_i\end{bmatrix},\ \hat{J} = \frac{\beta_i}{d_i}\frac{\partial f_{\mathrm{m}}}{\partial \hat{p}_i} - \delta_i.$$

Now, using Lemma 3, we get,

$$\bar{C}\bar{J}\bar{S} = \beta_i \frac{\partial}{\partial \theta_j}\left(m_i - \frac{f_m(\hat{y})}{d_i}\right) - (\delta_i + 1)\frac{\partial}{\partial \theta_j}\left(p_i - \hat{p}_i\right).$$

Also from the Lemma 3, for $\theta_j = \beta_i$, we have $\bar{C}\bar{Z} = m_i - \frac{f_m}{d_i}$ and

$$\dot{S}_e = \left(m_i - \frac{f_m}{d_i}\right) + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial \beta_i} - \delta_i \frac{\partial\left(p_i - \hat{p}_i\right)}{\partial \beta_i},$$

$$\Rightarrow \dot{S}_e = \left(m_i - \frac{f_m}{d_i}\right) + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial \beta_i} - \delta_i S_e. \tag{3}$$

For $\theta_j = \delta_i$, we have, $\bar{C}\bar{Z} = -(p_i - \hat{p}_i)$,

$$\dot{S}_e = -(p_i - \hat{p}_i) + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial \delta_i} - \delta_i \frac{\partial\left(p_i - \hat{p}_i\right)}{\partial \delta_i},$$

$$\Rightarrow \dot{S}_e = -(p_i - \hat{p}_i) + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial \delta_i} - \delta_i S_e. \tag{4}$$

For $\theta_j = d_i$, $\bar{C}\bar{Z} = \frac{\beta_i f_m}{d_i^2}$,

$$\dot{S}_e = \frac{\beta_i f_m}{d_i^2} + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial d_i} - \delta_i \frac{\partial\left(p_i - \hat{p}_i\right)}{\partial d_i},$$

$$\Rightarrow \dot{S}_e = \frac{\beta_i f_m}{d_i^2} + \beta_i \frac{\partial\left(m_i - \frac{f_m}{d_i}\right)}{\partial d_i} - \delta_i S_e. \tag{5}$$

Hence, using equations (3), (4) and (5), the sensitivity of the error in model reduction converges to zero if,

$$m_i = \frac{f_m(\hat{y})}{d_i} \Rightarrow m_T = \frac{K b_T^n}{d_T(b_T^n + p_L^n)},\ m_L = \frac{K b_L^n}{d_L(b_L^n + p_T^n)}, \tag{6}$$

and trivially, the other condition is that the outputs are alike $y = \hat{y}$.

We run the full model in equation (2) through the automated model reduction algorithm with the desired tolerance levels for the error and the sensitivity of the error. The reduced model returned by our algorithm is given below. It is evident that in this case the reduced model is the same as that given by the singular perturbation theory:

$$\dot{\hat{p}}_T = \frac{\beta_T K b_T^n}{d_T(b_T^n + \hat{p}_L^n)} - \delta_T \hat{p}_T$$
$$\dot{\hat{p}}_L = \frac{\beta_L K b_L^n}{d_L(b_L^n + \hat{p}_T^n)} - \delta_L \hat{p}_L.$$

However, if we tune the error margins and desire lower tolerance levels then the algorithm returns a different reduced model which would be one with additional states. It would be difficult to derive such a reduced model using singular perturbation approach. Hence, we can use the automated model reduction algorithm to control the tradeoff between the number of states and error in model reduction along with the sensitivity of the error to find reduced models that would be otherwise elusive to find using other existing methods. We will see with the next example that minimizing the error in reduction is not enough as it returns a set of reduced models that all have similar error bounds. We will use the automated model reduction algorithm to find out the reduced model that has the best robust performance in presence of parametric uncertainties to choose one final reduced model out of the multiple possibilities we might have when minimizing only the error.

B. Population and composition control circuit

In this example, we consider a synthetic circuit designed to control a consortium of two different cell strains. Two coupled feedback controllers are designed using various components that control the total population density of the consortium to a desired value and to control the fraction of the two cell types in the consortium. In [30], the theory behind the design of these controllers has been discussed and [31] discusses the details of the implementation of this circuit. In this circuit, each cell in the consortium promotes the production of a toxin that kills itself. At the same time, each cell also signals the production of anti-toxin in the cells of the other type using signaling molecules that can transport over from one cell to other. The production of anti-toxin rescues the cell from killing itself by binding the toxin in that cell. Amount of two different inducers can be used to control the expression of these signaling molecules. There are two different fluorescent output readouts corresponding to each cell type. Hence, this system has two outputs (C_1 and C_2 in the model).

1) Mathematical model: The model dynamics can be described by the ODE model given in equation (7). The description of the model species and the model parameters are given in Appendix B. For more details on the parameter values and description, the reader is referred to [31].

$$\begin{aligned}
\frac{dT_1}{dt} &= \beta_{S_1}\left(l_{S_1} + \frac{S_1^2}{K_{S_1} + S_1^2}\right) - k_b A_1 T_1 - d_T T_1 \\
\frac{dA_1}{dt} &= K_r \beta_{S_2}\left(l_{S_2} + \frac{S_2^2}{K_{S_2} + S_2^2}\right) - k_b A_1 T_1 - d_T A_1 \\
\frac{dS_1}{dt} &= \beta_{tac}\left(l_{tac} + \frac{I^2}{K_{tac} + I^2}\right) C_1 - d_S S_1 \\
\frac{dS_2}{dt} &= \beta_{sal}\left(l_{sal} + \frac{L^2}{K_{sal} + L^2}\right) C_2 - d_S S_2 \\
\frac{dT_2}{dt} &= \beta_{S_2}\left(l_{S_2} + \frac{S_2^2}{K_{S_2} + S_2^2}\right) - k_b A_2 T_2 - d_T T_2 \\
\frac{dA_2}{dt} &= K_r \beta_{S_1}\left(l_{S_1} + \frac{S_1^2}{K_{S_1} + S_1^2}\right) - k_b A_2 T_2 - d_T A_2 \\
\frac{dC_1}{dt} &= k_C\left(1 - \frac{C_1 + C_2}{C_{max}}\right) C_1 - d_c C_1 \frac{T_1}{K_{tox} + T_1} - dC_1 \\
\frac{dC_2}{dt} &= k_C\left(1 - \frac{C_1 + C_2}{C_{max}}\right) C_2 - d_c C_2 \frac{T_2}{K_{tox} + T_2} - dC_2.
\end{aligned} \tag{7}$$

2) Model reduction: For the model of this system, using singular perturbation theory to reduce the model is not possible as it is not clear how this model can be expressed in the singular perturbation framework as defined in Theorem 1. We will show how the automated model reduction developed earlier can be used to reduce the model to get a reduced model that has 4 states, down from a total of 8 in the full model. As discussed earlier, this model reduction could be beneficial in both the design guide for the circuit and the analysis using the model. With the four state reduced model, we can make easier and direct connections with the tunable parameters in the system since there are only two inputs and two outputs in the system. Using simulations, we can study the response of the system for different parameters and make connections with the experimental setup to guide the design of the experiment. Moreover, the control functions implemented in this circuit can be conceptually achieved in multiple ways. For instance, in the circuit described above, both cells kill self and rescue the other. There are other possible configurations that could implement similar control functions in principle (repressing growth/promoting death of the other cell etc.), we need to scan over all such possibilities to determine which one could be implemented experimentally to give the desired results. Using reduced models for such purpose is important as working

with complicated models can be disconnected from the control functions whereas with a reduced model this is not the case as much. The advantages of using a reduced model for parameter identification are clear, since, we only have two measurements and 24 parameters in the full model. However, as we will see in the final reduced model there are only 4 states and 13 parameters. Hence, the number of non-identifiable parameters will be much lesser, giving better estimation of parameters. Using the automated model reduction algorithm for this model, we get four different reduced order model (all with 4 states) that perform similar on the normed error metric. The results are presented in Fig. 2. Running the algorithm with both

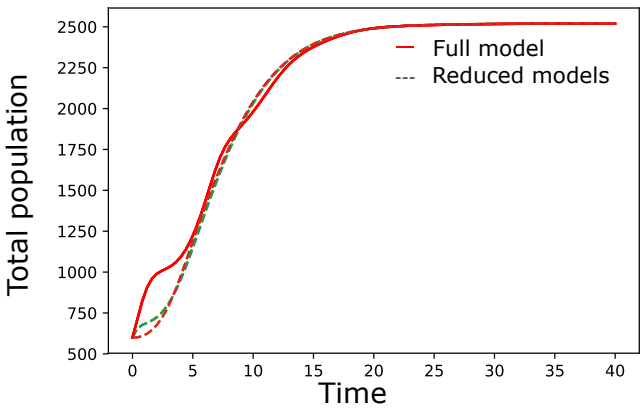


Fig. 2. Control of total population density of the consortium to a desired level compared for the full model and the different reduced models obtained when only considering the normed error metric. It is clear that all of the four models have similar performance and so it is not clear from this metric what the final reduced model choice should be. Note that the total population response of two of the reduced models are stacked on top of others in this figure.

the minimization of sensitivity of error metric (Fig. 3) and the minimization of the normed error, we can find out the following reduced model that has robust performance in total population density control.

$$
\begin{aligned}
\hat{f}_1 &= \left(l_{S1} + \frac{\beta_{S1} x_7^2}{x_7^2 + K_{I0}}\right) - d_T x_1 \\
&\quad - \frac{\beta_{S2} k_b x_1 \left(K_{a0} l_{S2} + x_8^2\right)}{k_b x_1 x_8^2 + K_{a0} k_b x_1 + d_T x_8^2 + K_{a0} d_T}, \\
\hat{f}_2 &= \left(l_{S2} + \frac{\beta_{S2} x_8^2}{x_7^2 + K_{a0}}\right) - d_T x_5 \\
&\quad - \frac{\beta_{S1} k_b x_5 \left(K_{I0} l_{S1} + x_7^2\right)}{k_b x_5 x_7^2 + K_{I0} k_b x_5 + d_T x_7^2 + K_{I0} d_T}, \\
\hat{f}_3 &= k_c \left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_7 - \frac{d_c x_1 x_7}{x_1 + K_{tox}} - d x_7, \qquad (8) \\
\hat{f}_4 &= k_c \left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_8 - \frac{d_c x_5 x_8}{x_5 + K_{tox}} - d x_8.
\end{aligned}
$$

Here $\hat{x} = \begin{bmatrix} x_1 & x_5 & x_7 & x_8 \end{bmatrix}^T$ and the lumped parameters are derived in Appendix C.

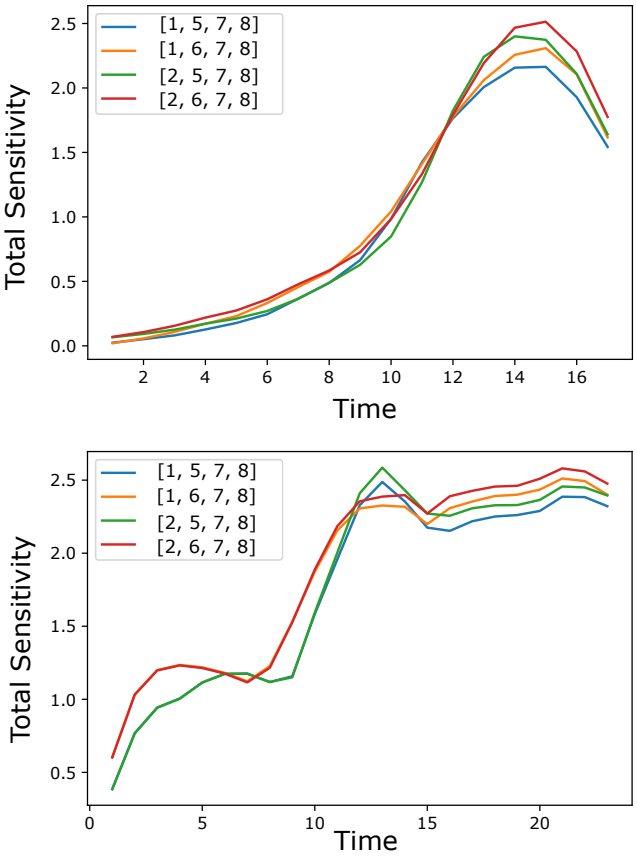


Fig. 3. (top) The total sensitivity of output y_1, that corresponds to the population of C_1, with time for all four reduced models. The legend indicates the state indices that form the reduced model state vector $\hat{x}$. (bottom) The total sensitivity of output y_2, that corresponds to the total population of C_2, with time for all four reduced models. Observe that for both outputs we see that the reduced model with $\hat{x} = \begin{bmatrix} x_1 & x_5 & x_7 & x_8 \end{bmatrix}^T$ has the lowest total sensitivity. Note that the total sensitivity of an output is the sum of the sensitivities of the output with respect to all model parameters. For detailed sensitivity analysis, refer to the Appendix C.

V. Conclusion

We proposed an algorithm based on quasi-steady state approximation model reduction that collapses the dynamics of a subset of states of the full model to algebraic relationships. Motivated by applications to reduce synthetic biological circuit models, we developed a structured model reduction algorithm that minimizes the error and its sensitivity between the full model and the reduced model. The algorithm returns a reduced model that not only has similar output responses as the full model but also is robust to parametric uncertainties. Towards that end, we extend the results of [28] by including sensitivity of error in the objective cost to search for reduced models. We show using an example of a toggle switch circuit how this additional constraint relates to the singular perturbation theory based model reduction. Moreover, using an example of a circuit designed to control the population and composition of a consortium consisting of two cell strains we show that error minimization as the only constraint does not give satisfactory

results in finding a reduced order model that best represents the full model. For this example, we show the application of our algorithm that minimizes the sensitivity of the error along with the error itself to return a final reduced model that has robust performance to parametric uncertainties. We also give theoretical results that bound the sensitivity of the error both for autonomous linear and nonlinear dynamics with linear output relationships. A direct extension of this work would be to derive results for more general system dynamics with inputs and nonlinear output relationships. It would also be interesting to look for ways to improve the computational efficiency of our algorithm, both by improving the theoretical results and by improving the algorithmic implementation.

VI. Acknowledgements

We would like to thank Chelsea Hu, Reed McCardell, and Shailja for insightful discussions. We would also like to thank Samuel Clamons for helping with sensitivity analysis computations. The author A.P. is supported by the Defense Advanced Research Projects Agency (Agreement HR0011-17-2-0008). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Domitilla Del Vecchio and Richard M Murray. *Biomolecular Feedback Systems*. Princeton University Press Princeton, NJ, 2015.

[2] Shev MacNamara, Kevin Burrage, and Roger B Sidje. "Multiscale modeling of chemical kinetics via the master equation". In: *Multiscale Modeling & Simulation* 6.4 (2008), pp. 1146–1168.

[3] Vassilios Sotiropoulos et al. "Model reduction of multiscale chemical Langevin equations: a numerical case study". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6.3 (2009), pp. 470–482.

[4] Darren J Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC, 2006.

[5] Petar V Kokotovic, Robert E O'Malley Jr, and Peddapullaiah Sannuti. "Singular perturbations and order reduction in control theory - an overview". In: *Automatica* 12.2 (1976), pp. 123–132.

[6] Bruce Moore. "Principal component analysis in linear systems: Controllability, observability, and model reduction". In: *IEEE Transactions on Automatic Control* 26.1 (1981), pp. 17–32.

[7] Dale F Enns. "Model reduction with balanced realizations: An error bound and a frequency weighted generalization". In: *The 23rd IEEE Conference on Decision and Control*. 1984, pp. 127–132.

[8] James Anderson, Yo-Cheng Chang, and Antonis Papachristodoulou. "Model decomposition and reduction tools for large-scale networks in systems biology". In: *Automatica* 47.6 (2011), pp. 1165–1174.

[9] Yi Liu and Brian DO Anderson. "Singular perturbation approximation of balanced systems". In: *International Journal of Control* 50.4 (1989), pp. 1379–1405.

[10] Daniela Degenring et al. "Sensitivity analysis for the reduction of complex metabolism models". In: *Journal of Process Control* 14.7 (2004), pp. 729–745.

[11] Ilse Smets et al. "Sensitivity function-based model reduction: A bacterial gene expression case study". In: *Biotechnology and bioengineering* 80.2 (2002), pp. 195–200.

[12] Mano R. Maurya et al. "Mixed-integer nonlinear optimisation approach to coarse-graining biochemical networks". In: *IET systems biology* 3.1 (2009), pp. 24–39.

[13] Katalin M Hangos, Attila Gábor, and Gábor Szederkényi. "Model reduction in bio-chemical reaction networks with Michaelis-Menten kinetics". In: *2013 European Control Conference (ECC)*. IEEE. 2013, pp. 4478–4483.

[14] Aristides Dokoumetzidis and Leon Aarons. "Proper lumping in systems biology models". In: *IET systems biology* 3.1 (2009), pp. 40–51.

[15] James D. Murray. "Mathematical Biology. I. An Introduction." In: *Photosynthetica* 40.3 (2002), pp. 414–414.

[16] Thomas J Snowden, Piet H Van Der Graaf, and Marcus J Tindall. "A combined model reduction algorithm for controlled biochemical systems". In: *BMC systems biology* 11.1 (2017), p. 17.

[17] Ovidiu Radulescu et al. "Reduction of dynamical biochemical reactions networks in computational biology". In: *Frontiers in genetics* 3 (2012), p. 131.

[18] Thomas J Snowden, Piet H van der Graaf, and Marcus J Tindall. "Methods of model reduction for large-scale biological systems: a survey of current methods and trends". In: *Bulletin of mathematical biology* 79.7 (2017), pp. 1449–1486.

[19] Edda Klipp et al. *Systems Biology: A Textbook*. John Wiley & Sons, 2016.

[20] Ulrich Maas and Stephen B Pope. "Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds". In: *Symposium (International) on Combustion*. Vol. 24. 1. Elsevier. 1992, pp. 103–112.

[21] Irina Surovtsova et al. "Accessible methods for the dynamic time-scale decomposition of biochemical systems". In: *Bioinformatics* 25.21 (2009), pp. 2816–2823.

[22] Klaus R Schneider and Thomas Wilhelm. "Model reduction by extended quasi-steady-state approximation". In: *Journal of mathematical biology* 40.5 (2000), pp. 443–450.

[23] Nishith Vora and Prodromos Daoutidis. "Nonlinear model reduction of chemical reaction systems". In: *AIChE Journal* 47.10 (2001), pp. 2320–2332.

[24] Friedrich G Helfferich. "Systematic approach to elucidation of multistep reaction networks". In: *The Journal of Physical Chemistry* 93.18 (1989), pp. 6676–6681.

[25] Johannes A. Christiansen. "The elucidation of reaction mechanisms by the method of intermediates in quasi-

stationary concentrations". In: *Advances in Catalysis*. Vol. 5. Elsevier, 1953, pp. 311–353.

[26] Hassan K Khalil and Jessy W Grizzle. *Nonlinear Systems*. Vol. 3. Prentice hall Upper Saddle River, NJ, 2002.

[27] Robert P Dickinson and Robert J Gelinas. "Sensitivity analysis of ordinary differential equation systems - a direct method". In: *Journal of computational physics* 21.2 (1976), pp. 123–143.

[28] Antonis Papachristodoulou et al. "Structured model reduction for dynamical networked systems". In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE. 2010, pp. 2670–2675.

[29] Antonis Papachristodoulou et al. "SOSTOOLS version 3.00 sum of squares optimization toolbox for MATLAB". In: *arXiv preprint arXiv:1310.4716* (2013).

[30] Xinying Ren et al. "Population regulation in microbial consortia using dual feedback control". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 5341–5347.

[31] Reed D. McCardell, Ayush Pandey, and Richard M. Murray. "Control of density and composition in an engineered two-member bacterial community". In: *bioRxiv* (2019). DOI: 10.1101/632174.

APPENDIX

A. *Proof of Theorem 4*

Recall that

$$\dot{\bar{S}} = \bar{J}\bar{S} + \bar{Z},$$
$$S_e = \bar{C}\bar{S}.$$

Assume that we want to bound the norm of the sensitivity of error for a finite time until which the system reaches steady state. Let us denote the final time by N. It can be chosen to be a time after which the system states have reached their equilibrium values. We have

$$\|S_e\|_2^2 = \int_0^N \bar{S}\bar{C}^T\bar{C}\bar{S}dt.$$

Given that there exists a $P = P^T \succeq 0$ such that $\bar{J}^T P + P\bar{J} = -\bar{C}^T\bar{C}$, consider a function $V(\bar{S}) = \bar{S}^T P\bar{S}$. Differentiating this function with respect to time, we have using Lemma 3,

$$\frac{dV}{dt} = \bar{S}^T(\bar{J}^T P + P\bar{J})\bar{S} + (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}).$$

Now, integrating the expression from 0 to N and then substituting the expression for $\|S_e\|_2^2$,

$$\|S_e\|_2^2 = \int_0^N -\frac{dV}{dt}dt + \int_0^N (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt,$$
$$\|S_e\|_2^2 = -V(\bar{S}_N) + V(\bar{S}_0) + \int_0^N (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt,$$

Since, P is a positive semidefinite matrix, V will be a non-negative function. Using this fact, we have the inequality

$$\|S_e\|_2^2 \leq \bar{S}_0^T P\bar{S}_0 + \int_0^N (\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z})dt. \tag{9}$$

For the first part of this equation, since at time zero the sensitivity coefficients are all zero or equal to 1 when the initial condition is dependent on parameters, we can write

$$\left|\bar{S}_0^T P\bar{S}_0\right| \leq \lambda_{\max}(P) \tag{10}$$

where we used We will now evaluate the second part in equation (9),

$$\begin{aligned}\int_0^N \left(\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}\right) dt &\leq \left\| \int_0^N \left(\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}\right) dt \right\|_2 \\ &\leq \int_0^N \left\| \left(\bar{Z}^T P\bar{S} + \bar{S}^T P\bar{Z}\right) \right\|_2 dt \\ &\leq 2N \max_t \left\| \bar{Z}^T P\bar{S} \right\|_2 . \end{aligned} \tag{11}$$

Recall that we have

$$x = T\begin{bmatrix}\hat{x} \\ x_c\end{bmatrix}, \tag{12}$$

where T is the permutation matrix, $\hat{x}$ is the state vector for the reduced model and x_c is the vector of all collapsed states as defined in Section III. Using the definition of the augmented sensitivity system from Lemma (3), we can now write

$$\bar{S} = \begin{bmatrix}S \\ \hat{S}\end{bmatrix} = \begin{bmatrix}T_1\frac{\partial\hat{x}}{\partial\theta} + T_2\frac{\partial x_c}{\partial\theta} \\ \frac{\partial\hat{x}}{\partial\theta}\end{bmatrix}, \bar{Z} = \begin{bmatrix}\frac{\partial f}{\partial\theta} \\ \frac{\partial\hat{f}}{\partial\theta}\end{bmatrix}.$$

Substituting in equation (11),

$$\begin{aligned}\bar{Z}^T P\bar{S} &= \begin{bmatrix}\frac{\partial f}{\partial\theta} \\ \frac{\partial\hat{f}}{\partial\theta}\end{bmatrix}^T \begin{bmatrix}P_{11}T_1\frac{\partial\hat{x}}{\partial\theta} + P_{11}T_2\frac{\partial x_c}{\partial\theta} + P_{12}\frac{\partial\hat{x}}{\partial\theta} \\ P_{21}T_1\frac{\partial\hat{x}}{\partial\theta} + P_{21}T_2\frac{\partial x_c}{\partial\theta} + P_{22}\frac{\partial\hat{x}}{\partial\theta}\end{bmatrix}, \\ &= \begin{bmatrix}\frac{\partial f}{\partial\theta} \\ \frac{\partial\hat{f}}{\partial\theta}\end{bmatrix}^T \begin{bmatrix}P_{11}T_1 + P_{12} & P_{11}T_2 \\ P_{21}T_1 + P_{22} & P_{21}T_2\end{bmatrix} \begin{bmatrix}\frac{\partial\hat{x}}{\partial\theta} \\ \frac{\partial x_c}{\partial\theta}\end{bmatrix}.\end{aligned}$$

Hence,

$$\int_0^N \left(\bar{Z}^T P \bar{S} + \bar{S}^T P \bar{Z}\right) dt \leq 2N \max_t \left\| \begin{bmatrix} \frac{\partial f}{\partial \theta} \\ \frac{\partial \hat{f}}{\partial \theta} \end{bmatrix}^T Q_s \begin{bmatrix} \frac{\partial \hat{x}}{\partial \theta} \\ \frac{\partial x_c}{\partial \theta} \end{bmatrix} \right\|_2,$$

where Q_s is as defined in the Theorem. Combining the two parts and substituting in equation (9), we get the desired result. Substituting $f = Ax$ and $\hat{f} = \hat{A}\hat{x}$, we can prove Theorem 3 by using equation 12 and regrouping terms as desired.

B. Population and composition control circuit details

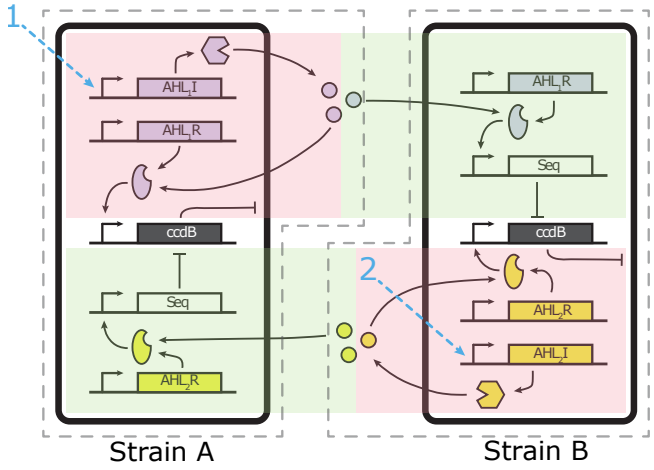


Fig. 4. Schematic of the composition and population density control synthetic biological circuit of a two-member bacterial community. Figure and caption adapted from [31] with permission. It is a symmetric circuit motif in its two cells to create cis-acting negative feedback loops on each member and trans acting rescues from negative feedback from each member to the other. Negative feedback and sequestration rescue are effected by ccdB toxin (T_i in the model) and its ccdA antitoxin (A_i), respectively. Signals S_1 and S_2 are chemical inducers that activate transcription of AHL synthase genes. The observations are fluorescent outputs corresponding to the population of each cell strain C_1 and C_2. Note here that the Strain A corresponds to C_1 and Strain B to C_2.

TABLE I
MODEL PARAMETERS

S.no.	Parameters	Description	Unit	Guess
1	β_{S_1}	Max transcription rate of Lux inducible promoter	con./hr	6
2	l_{S_1}	Leak constant of Lux inducible promoter	N/A	2e-3
3	K_{S_1}	Activation constant of Lux inducible promoter	con.	430
4	k_b	Binding rate between toxin and anti-toxin	1/con.hr	30
5	β_{S_2}	Max transcription rate of Cin inducible promoter	con./hr	6
6	l_{S_2}	Leak constant of Cin inducible promoter	N/A	2e-3
7	K_{S_2}	Activation constant of Cin inducible promoter	con.	30
8	β_{tac}	Max transcription rate of inducible promoter P_{tac}	con./hr	19.8e-3
9	l_{tac}	Leak constant of inducible promoter P_{tac}	N/A	1.5e-3
10	K_{tac}	Activation constant of inducible promoter P_{tac}	con.	1.4e5
11	β_{sal}	Max transcription rate of inducible promoter P_{sal}	con./hr	14.4e-3
12	l_{sal}	Leak constant of inducible promoter P_{sal}	N/A	2.1e-4
13	K_{sal}	Activation constant of inducible promoter P_{sal}	con.	4.3e4
14	k_C	Cell Division rate	1/hr	0.6
15	C_{max}	Population cap	conc.	5500
16	d_c	Cell death rate	1/conc.hr	0.8
17	S_2	Max induced Cin concentration	con.	100
18	I	Max induced IPTG concentration	con.	1e6
19	S_1	Max induced Lux concentration	con.	1e3
20	Sal	Max induced Sal concentration	con.	1e5
21	K_{tox}	Repression coefficient of toxin to proliferation	con.	5
22	d_S	Degradation constant of AHLs	1/hr	0.1
23	d	Basal degradation of each cell	1/hr	0.01
24	d_T	Basal degradation of toxins and antitoxins	1/hr	0.01
25	K_r	Ribosome scaling factor	N/A	5

TABLE II
MODEL SPECIES

Species	Description
T_1	Average toxin (ccdB) con. in C_1 population
A_1	Average anti-toxin (ccdA) con. in C_1 population
S_1	Signal 1 (S_1), Lux con. in consortia
S_2	Signal 2 (S_2), Cin con. in consortia
T_2	Average toxin (ccdB) con. in C_2 population
A_2	Average anti-toxin (ccdA) con. in C_2 population
C_1	Cell type 1 (C_1) population count
C_2	Cell type 2 (C_2) population count

C. Model reduction of population control circuit

When the dynamics for x_2, x_3, x_4, x_6 are collapsed into algebraic relationships, i.e. $\hat{x} = \begin{bmatrix} x_1 & x_5 & x_7 & x_8 \end{bmatrix}^T$, we get the following reduced order model,

$$\hat{f}_1 = \beta_{S1}\left(\frac{\beta_{lac}^2 x_7^2\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2}{d_S^2\left(I^2 + K_{lac}\right)^2\left(K_{S1} + \frac{\beta_{lac}^2 x_7^2}{d_S^2 (I^2+K_{lac})^2}\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2\right)} + l_{S1}\right)$$

$$- \frac{\beta_{S2} k_b x_1\left(K_{S2} l_{S2} + \frac{\beta_{tet}^2 l_{S2} x_8^2}{d_S^2 (K_{tet}+atc^2)^2}\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2 + \frac{\beta_{tet}^2 x_8^2}{d_S^2 (K_{tet}+atc^2)^2}\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2\right)}{K_{S2} d_T + K_{S2} k_b x_1 + \frac{\beta_{tet}^2 d_T x_8^2}{d_S^2 (K_{tet}+atc^2)^2}\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2 + \frac{\beta_{tet}^2 k_b x_1 x_8^2}{d_S^2 (K_{tet}+atc^2)^2}\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2} - d_T x_1$$

$$\hat{f}_2 = -\frac{\beta_{S1} k_b x_5\left(K_{S1} l_{S1} + \frac{\beta_{lac}^2 l_{S1} x_7^2}{d_S^2 (I^2+K_{lac})^2}\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2 + \frac{\beta_{lac}^2 x_7^2}{d_S^2 (I^2+K_{lac})^2}\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2\right)}{K_{S1} d_T + K_{S1} k_b x_5 + \frac{\beta_{lac}^2 d_T x_7^2}{d_S^2 (I^2+K_{lac})^2}\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2 + \frac{\beta_{lac}^2 k_b x_5 x_7^2}{d_S^2 (I^2+K_{lac})^2}\left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2}$$

$$+ \beta_{S2}\left(\frac{\beta_{tet}^2 x_8^2\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2}{d_S^2\left(K_{S2} + \frac{\beta_{tet}^2 x_8^2}{d_S^2 (K_{tet}+atc^2)^2}\left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2\right)\left(K_{tet} + atc^2\right)^2} + l_{S2}\right) - d_T x_5,$$

$$\hat{f}_3 = -d x_7 - \frac{d_c x_1 x_7}{K_{tox} + x_1} + k_c\left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_7,$$

$$\hat{f}_4 = -d x_8 - \frac{d_c x_5 x_8}{K_{tox} + x_5} + k_c\left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_8.$$

Lumping parameters together, we get

$$\hat{f}_1 = -d_T x_1 - \frac{\beta_{S2} k_b x_1\left(K_{a0} l_{S2} + x_8^2\right)}{k_b x_1 x_8^2 + K_{a0} k_b x_1 + d_T x_8^2 + K_{a0} d_T} + \left(l_{S1} + \frac{\beta_{S1} x_7^2}{x_7^2 + K_{I0}}\right),\ K_{ta} \triangleq \left(K_{tet} l_{tet} + atc^2 l_{tet} + atc^2\right)^2,$$

$$\hat{f}_2 = -d_T x_5 - \frac{\beta_{S1} k_b x_5\left(K_{I0} l_{S1} + x_7^2\right)}{k_b x_5 x_7^2 + K_{I0} k_b x_5 + d_T x_7^2 + K_{I0} d_T} + \left(l_{S2} + \frac{\beta_{S2} x_8^2}{x_7^2 + K_{a0}}\right),\ K_{lI} \triangleq \left(I^2 l_{lac} + I^2 + K_{lac} l_{lac}\right)^2,$$

$$\hat{f}_3 = -\frac{d_c x_1 x_7}{x_1 + K_{tox}} + k_c\left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_7 - d x_7, \qquad K_a \triangleq \left(K_{tet} + atc^2\right)^2, K_I \triangleq \left(K_{lac} + I^2\right)^2, K_{a0} \triangleq \frac{K_{S2}}{K_{\beta a}},$$

$$\hat{f}_4 = -\frac{d_c x_5 x_8}{x_5 + K_{tox}} + k_c\left(1 - \frac{x_7 + x_8}{C_{max}}\right) x_8 - d x_8, \qquad K_{\beta a} \triangleq \frac{\beta_{tet}^2 K_{ta}}{d_s^2 K_a}, K_{\beta I} \triangleq \frac{\beta_{lac}^2 K_{lI}}{d_s^2 K_I}, K_{I0} \triangleq \frac{K_{S1}}{K_{\beta I}}.$$

As we can see we have reduced the total number of parameters in the reduced model from a total of 24 in the full model to 13 in the reduced model. Other reduced models with four states are obtained by collapsing the dynamics for (x_2, x_3, x_4, x_5),

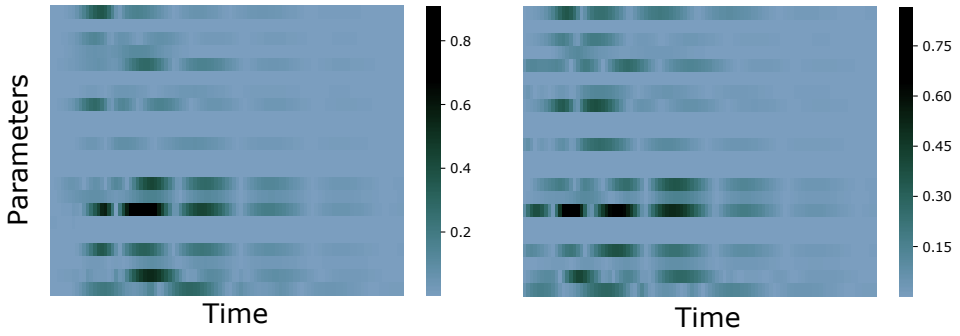


Fig. 5. Normalized sensitivity coefficients of the two outputs, y_1 (left) and y_2 (right) corresponding to x_7 and x_8, i.e. the population of each cell type, with respect to the model parameters for the reduced model shown above with $\hat{x} = \begin{bmatrix} x_1 & x_5 & x_7 & x_8 \end{bmatrix}^T$.

or (x_1, x_3, x_4, x_5), or (x_1, x_3, x_4, x_6). All four of these reduced models have four states and minimize the error metric. In Fig. 5, we show the sensitivity analysis heatmap for the two outputs of the reduced model derived above with respect to all model parameters. Similar heatmap plots can be obtained for all of the other reduced models, given in Fig. 6. Since it is hard to visually get a comparison of the sensitivities from these heatmaps, we used the norm of the total sensitivity of an output with respect to all parameters as in Fig. 3. Using this we then conclude that the reduced model in equation 8 has the lowest sensitivity out of all the others and hence has robust performance in the presence of parametric uncertainties.

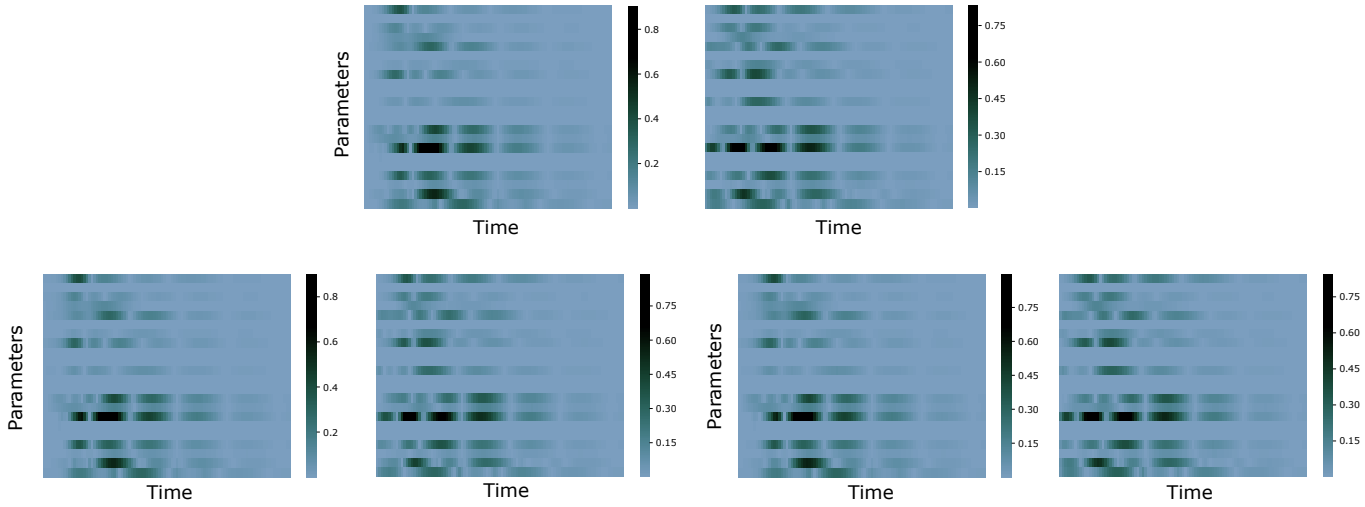
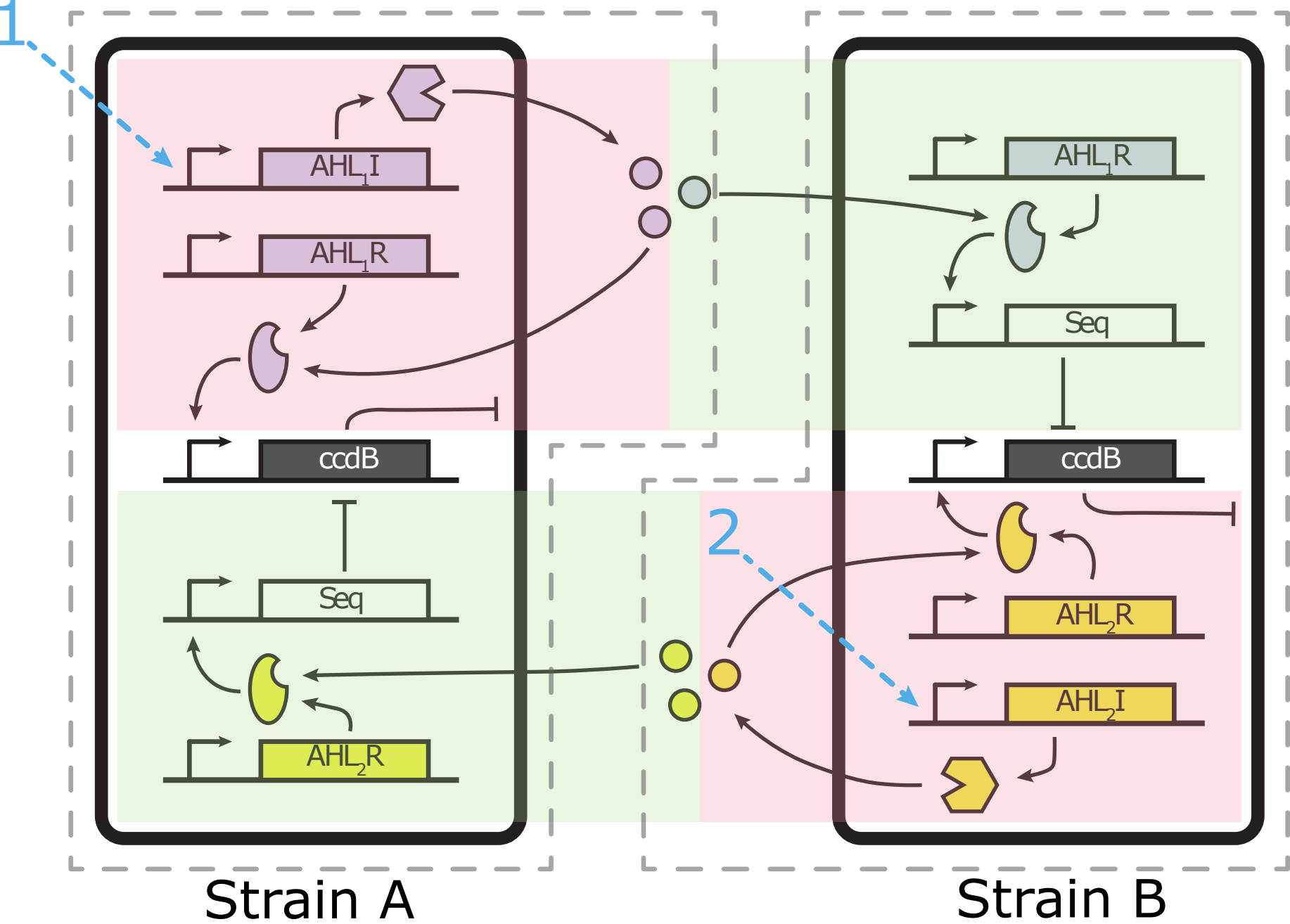
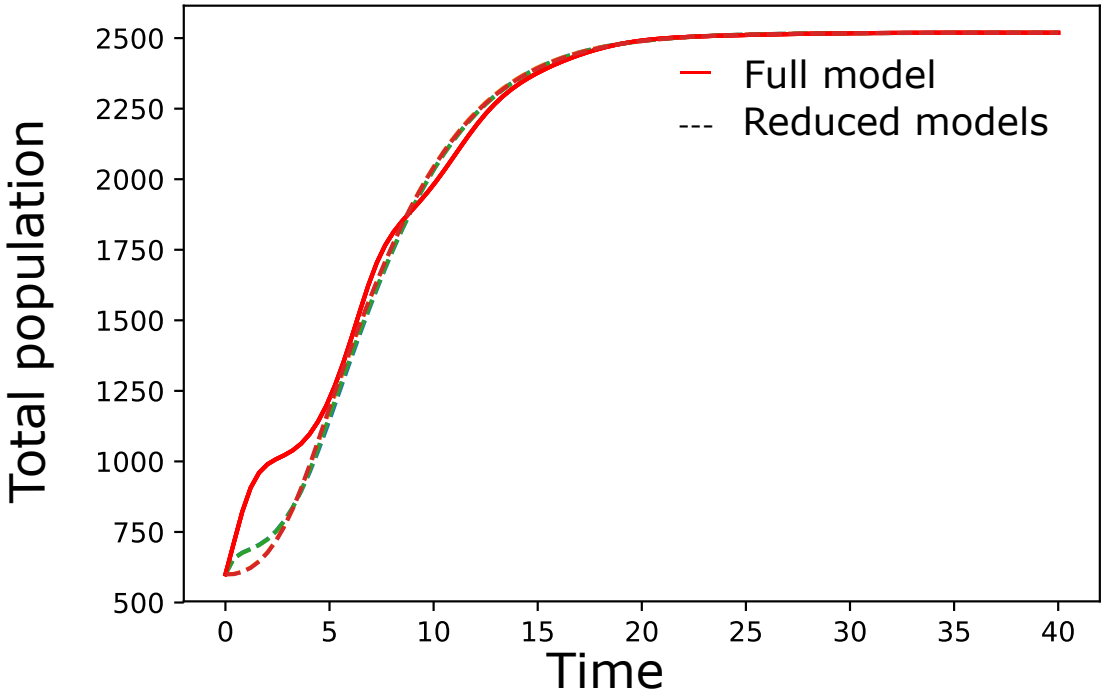
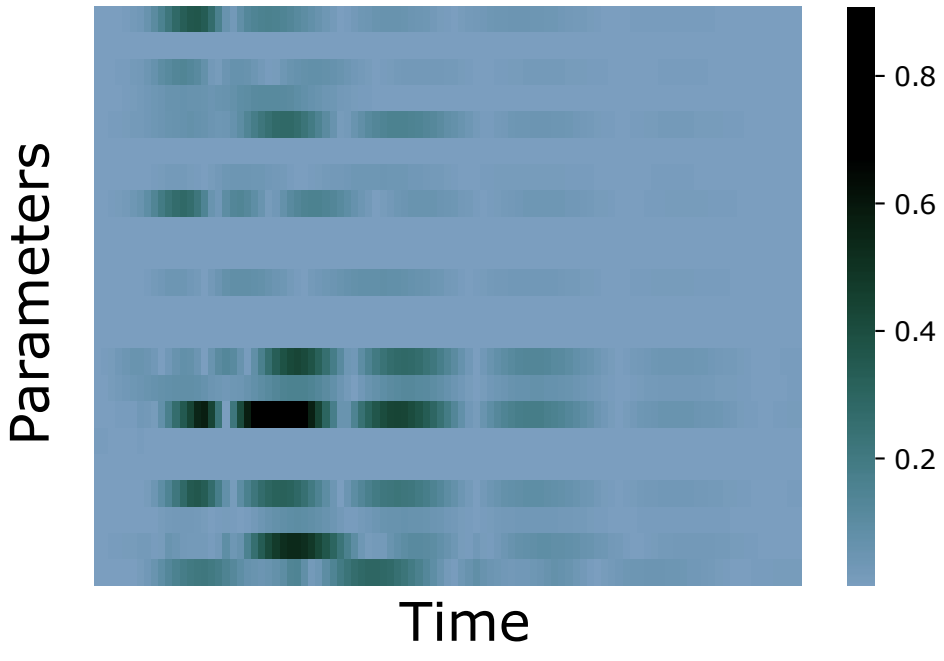
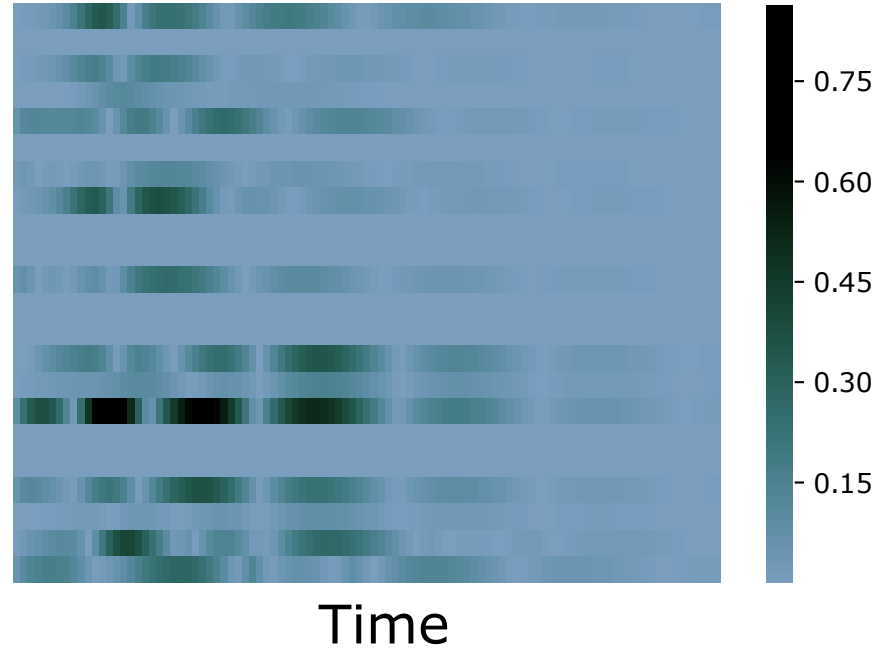


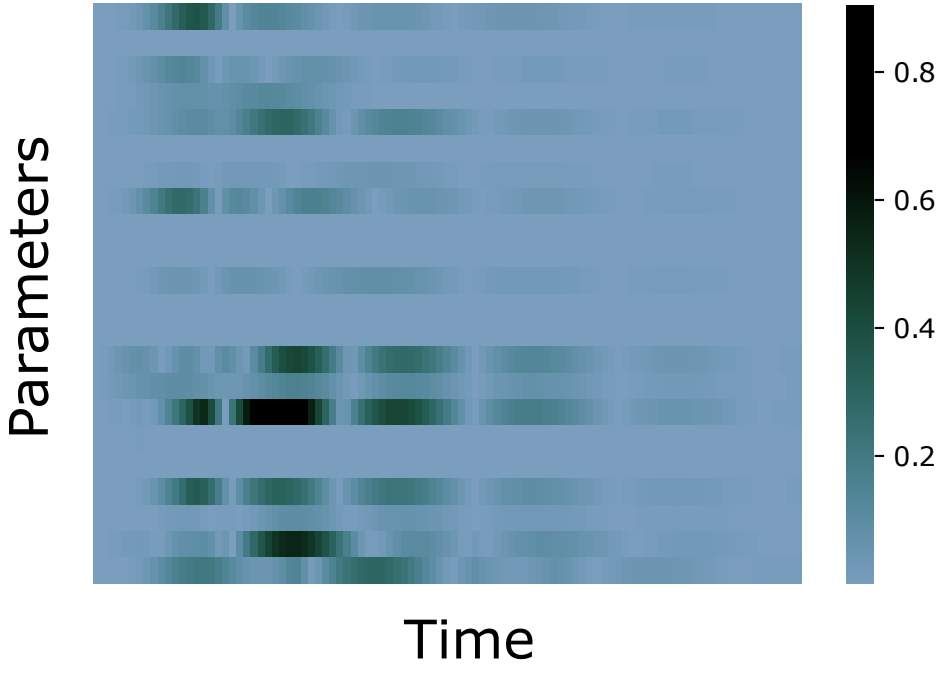
Fig. 6. Normalized sensitivity of the two outputs, y_1 and y_2 corresponding to x_7 and x_8, i.e. the population of each cell type, with respect to the model parameters for different reduced models. (top) We have the reduced model with $\hat{x} = \begin{bmatrix} x_1 & x_6 & x_7 & x_8 \end{bmatrix}^T$. (bottom left) We have the reduced model with with $\hat{x} = \begin{bmatrix} x_2 & x_5 & x_7 & x_8 \end{bmatrix}^T$ and (bottom right) we have $\hat{x} = \begin{bmatrix} x_2 & x_6 & x_7 & x_8 \end{bmatrix}^T$. Note that the left plot is the sensitivity analysis for the output y_1 and the right is for the second output y_2 in each case.

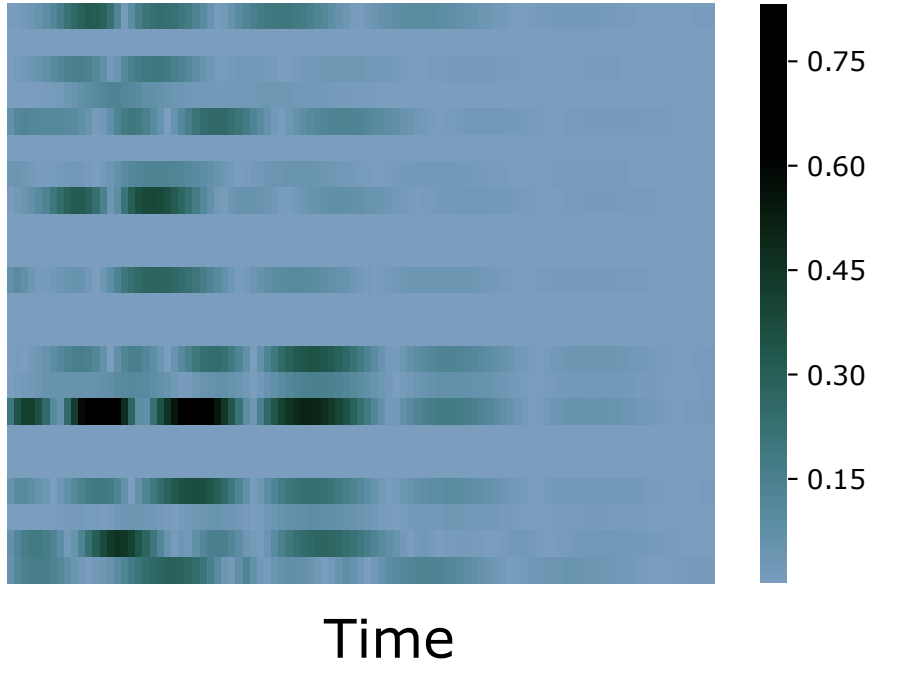

1
AHL1I
AHL1R
ccdB
Seq
AHL2R
2
AHL1R
Seq
ccdB
AHL2R
AHL2I
Strain A
Strain B

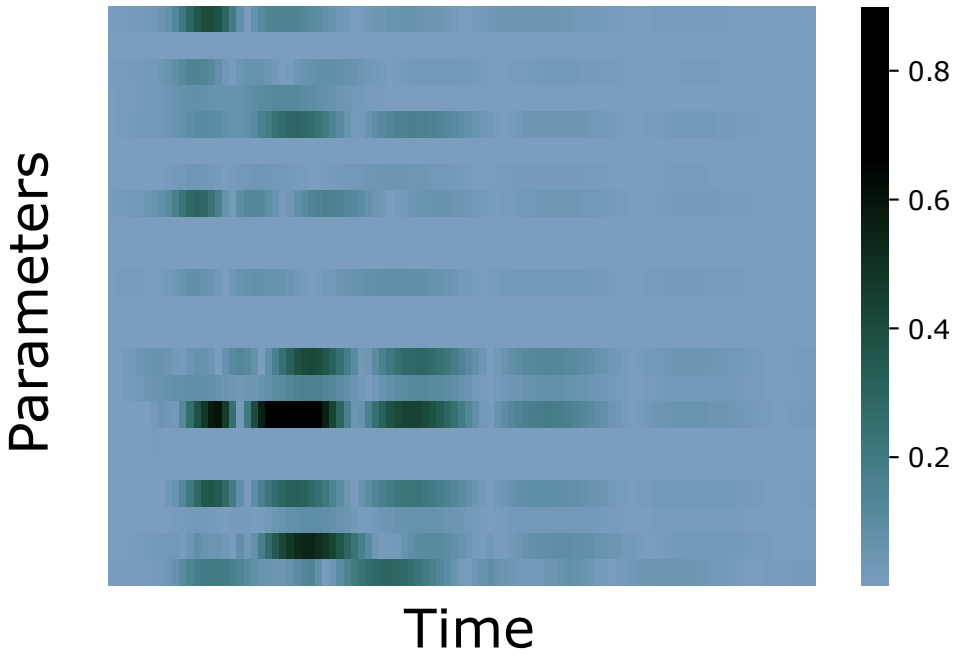

Full model
Reduced models
Total population
Time
2500
2250
2000
1750
1500
1250
1000
750
500
0
5
10
15
20
25
30
35
40

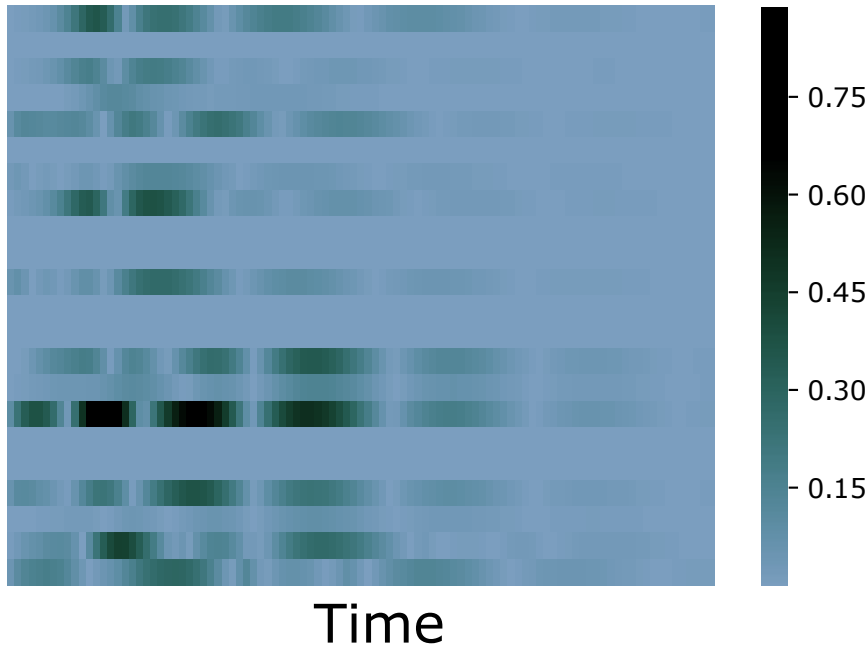

Parameters
Time
0.8
0.6
0.4
0.2

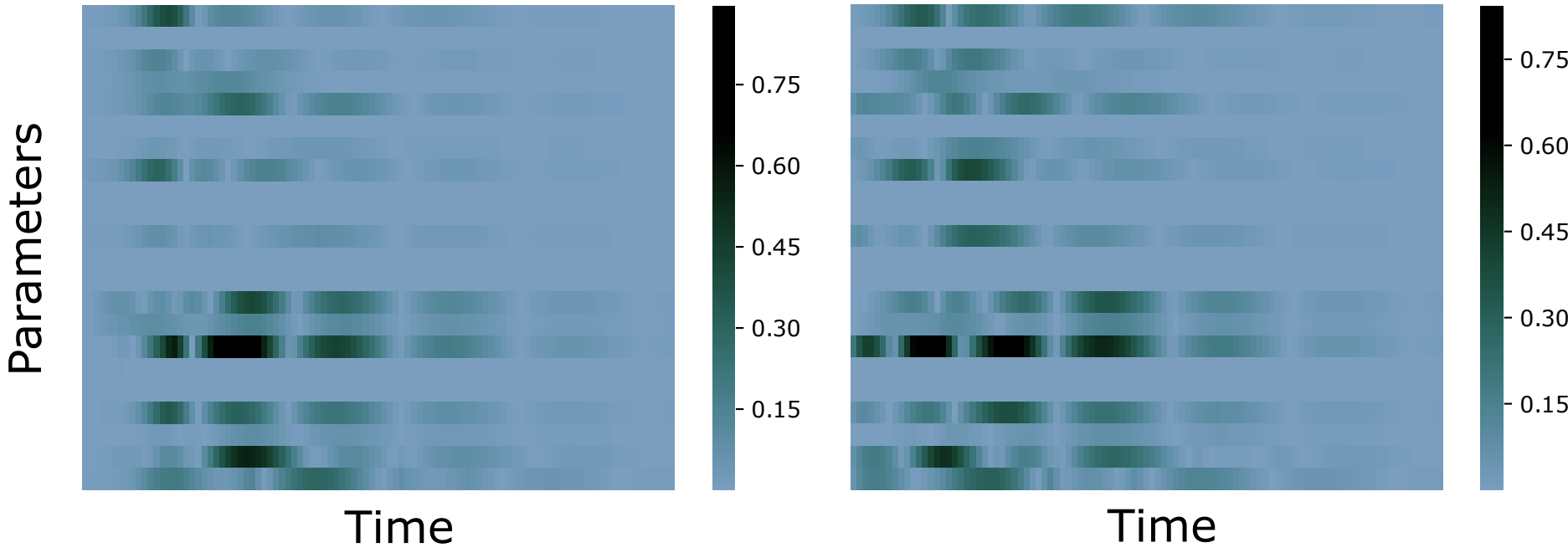

Time
0.75
0.60
0.45
0.30
0.15

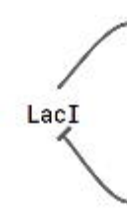

Parameters
0.8
0.6
0.4
0.2
Time

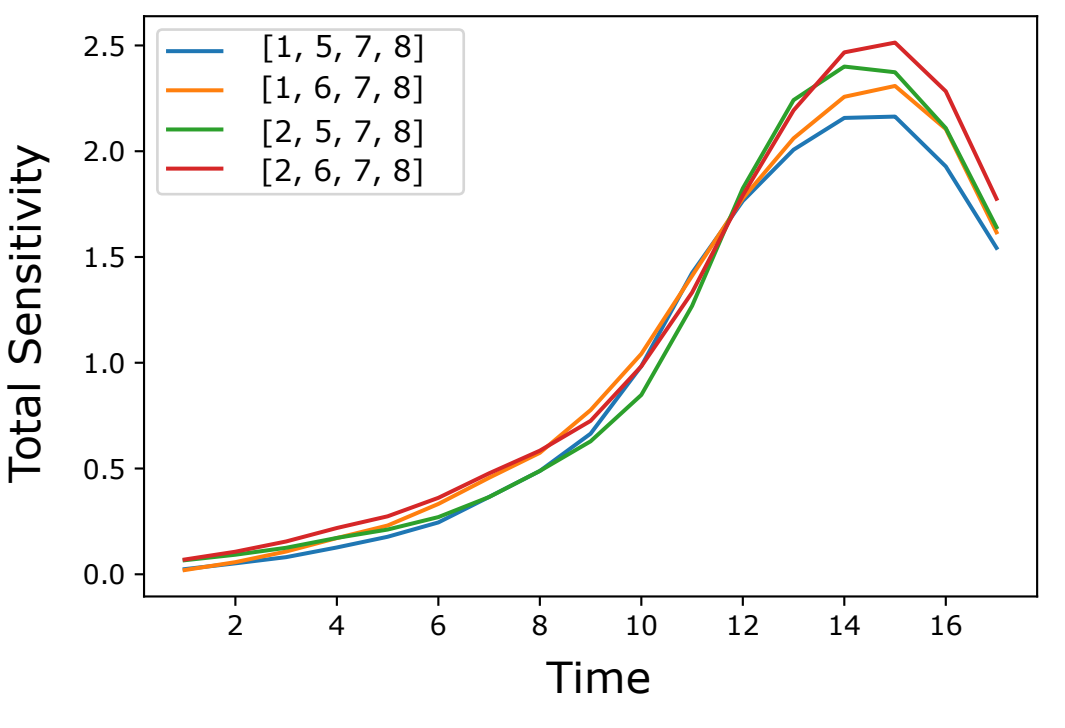

0.75
0.60
0.45
0.30
0.15
Time

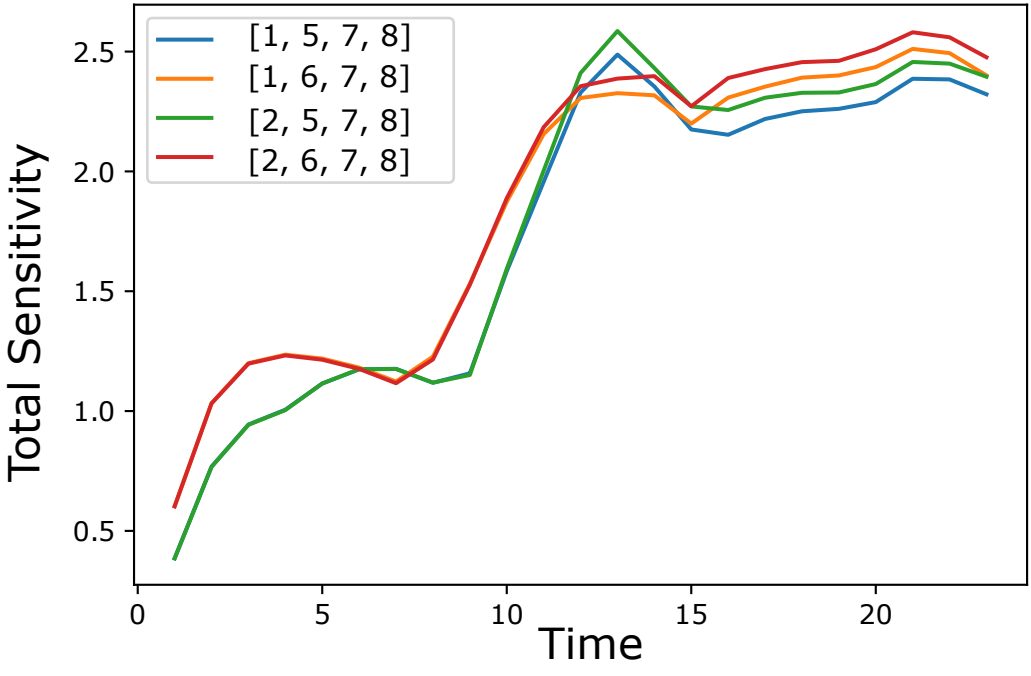

Parameters
0.8
0.6
0.4
0.2
Time


0.75
0.60
0.45
0.30
0.15
Time


Parameters
Time
0.75
0.60
0.45
0.30
0.15
Time
0.75
0.60
0.45
0.30
0.15


LacI
TetR


[1, 5, 7, 8]
[1, 6, 7, 8]
[2, 5, 7, 8]
[2, 6, 7, 8]
Total Sensitivity
Time
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
8
10
12
14
16


[1, 5, 7, 8]
[1, 6, 7, 8]
[2, 5, 7, 8]
[2, 6, 7, 8]
Total Sensitivity
Time
0.5
1.0
1.5
2.0
2.5
0
5
10
15
20