An Automatic Sparse Model Estimation Method Guided by Constraints That Encode System Properties

Zoltan A. Tuza[1] and Guy-Bart Stan[1]

***Abstract*— Finding efficient methods to estimate model parameters or build models from time-series data is a central quest in Systems and Synthetic Biology. To aid this search, existing parameter estimation methods were adapted from other fields, or new ones were developed. In this paper, we use the Sparse Bayesian Learning (SBL) framework, which was first developed in the field of signal processing, and implement it as an iterative convex optimization problem. We extend the existing SBL framework to accommodate constraints that enforce certain systems properties, such as nonnegative state variables or bounded state trajectories. These properties are vital parts of a dynamical model in biology and chemistry but are often overlooked in the parameter estimation literature. As a result of this work, the extended framework can automatically build "proper" dynamical models from time-series data. Finally, the examples show that such framework complemented with appropriate constraints can aid the model building process.**

I. INTRODUCTION

Estimating the parameters or the complete model structure of biological systems from time-series data is a crucial task in synthetic biology because it enables us to understand, analyse and optimise designs of synthetic biology constructs. However, building models manually for each data set is inconvenient and might become infeasible for highly complex synthetic systems.

To tackle this problem, our long-term goal is to develop a closed-loop system identification platform. The loop starts with high-throughput data acquisition, and the platform processes and aggregates the collected data into proper dynamical models automatically. Next, it tests the estimated models with the help of optimal experimental design [1], which guides the data acquisition in the next iteration. Finally, this procedure converges to an automatically built model that captures the underlying dynamics of the measured biological system.

Inside this platform, our modelling framework is within the class of kinetic systems, because it is capable of modelling the dynamics of the biological systems [2], and it has a well-defined chemical reaction network representation [3]. Moreover, the Ordinary Differential Equations (ODE) that describe the dynamics in this system class have a certain algebraic structure, and this property can be exploited during the system identification process. For example, a simple sign pattern of the right-hand side terms guarantees that the model remains in the system class [2]. These terms represent the reaction rates that can be either monomials (mass action kinetics) or rational functions (Hill or Michaelis-Menten kinetics) [4].

Two types of system identification problems can be formulated in this system class. First, the reaction rates are known, but not the corresponding reaction rate coefficients (or reaction rate constants). In this case, we need to estimate rate coefficients for each reaction rates. In the second case, only a set of possible reaction rates are known *a priori*. This set has many more terms than the model needs. The goal is then to identify which reaction rates are not required to describe the underlying biochemical process.

In both problems, most algorithms find the best fit to the data, but the underlying model might not be nonnegative or stable. Such solutions must be avoided by formulating appropriate constraints.

System identification procedures are widely reported in the literature [5], and the existing methods for parameter estimation of kinetic systems certainly benefit from this knowledge [6]. However the specificities of this system class—such as the algebraic structure or nonnegative state equations—are rarely exploited in these methods. The goal of this paper is to show how an automatic model building procedure, based on Sparse Bayesian Learning (SBL), can be complemented with constraints that enforce prescribed system properties. More importantly, other system identification approaches can use these constraints, as long as they fulfil certain conditions.

The paper is organized as follows: in Section II, we will introduce our modelling framework along with its algebraic structure; in Section III, we will report the sparse model building framework, which forms the backbone of our work. The constraints used to complement the SBL algorithm will be represented in Section IV. Finally, two examples illustrate the results in Section V.

Notations: $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers, $\mathbb{N}_0$ is the set of integers including zero. $[A]_{ij}$ denotes the entry in the ith row and jth column of the matrix A. Furthermore, $[A]_{i\cdot}$ denotes the ith row of matrix A.

II. KINETIC SYSTEMS

During the system identification procedure, we assume that the dynamics of the underlying biochemical processes can be modelled by the class of kinetic systems. In this system class, for example, mass action or rational kinetics (e.g. Hill or Michalis-Menten) can be represented.

Let us define the following ODE with polynomial right-hand side and with state vector $x \in \mathbb{R}^n_{\geq 0}$, as

$$\dot{x} = M\varphi(x), \quad x(0) \in \mathbb{R}^n_{\geq 0}, \tag{1}$$

[1]Imperial College London, Department of Bioengineering, SW7 2AZ London, UK. z.tuza@imperial.ac.uk, g.stan@imperial.ac.uk

where the matrix $M \in \mathbb{R}^{n\times m}$ is the coefficient matrix and $\varphi(x) : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^m_{\geq 0}$ is a monomial-type vector mapping defined as

$$\varphi_j(x) = \prod_{i=1}^{n} x_i^{[B]_{ij}}, j = 1, \ldots, m, \tag{2}$$

and $B \in \mathbb{N}_0^{n\times m}$.

Note that the system in (1), with certain sign constraints in matrix M, belongs to the class of nonnegative systems, i.e. $\mathbb{R}^n_{\geq 0}$ is forward invariant (see, e.g. Chapter 9 in [2] or in Section IV-A).

Next, we define Chemical Reaction Networks (CRNs), which can be characterized by three sets:

- a set of species: $\mathcal{S} = \{X_i \mid i = 1, \ldots, n\}$,
- a set of complexes: $\mathcal{C} = \{C_j \mid j = 1, \ldots, m\}$, where $C_j = \sum_{i=1}^{n} \alpha_{ji} X_i \qquad j = 1, \ldots, m$ and $\alpha_{ji} \in \mathbb{N}_0 \qquad j = 1, \ldots, m,\ i = 1, \ldots, n$, α_{ji} are called the stoichiometric coefficients,
- and a set of reactions: $\mathcal{R} \subseteq \{(C_i, C_j) \mid C_i, C_j \in \mathcal{C}\}$, each ordered pair (C_i, C_j) has a reaction rate coefficient $k_{ij} \in \mathbb{R}_{\geq 0}$ so that the corresponding reaction $C_i \to C_j$ takes place if and only if $k_{ij} > 0$.

In the rest of the paper, we assume mass action kinetics, but the results summarised in this section have been extended to rational kinetics as well, see e.g. [7] for more details.

For computation purposes, we can characterise a CRN by two matrices: the complex composition matrix $Y \in \mathbb{N}_0^{n\times m}$ describes the complexes as follows

$$[Y]_{ij} = \alpha_{ji} \qquad i = 1, \ldots, n,\ j = 1, \ldots, m,$$

and the set of reactions is encoded by the Kirchhoff matrix $A_\kappa \in \mathbb{R}^{m\times m}$ as

$$[A_\kappa]_{ij} = \begin{cases} k_{ji} & \text{if } i \neq j \\ -\sum\limits_{l=1, l\neq i}^{m} k_{il} & \text{if } i = j. \end{cases} \tag{3}$$

The dynamics of a CRN can be written as a nonnegative polynomial differential equation

$$\dot{x} = Y A_\kappa \psi^Y(x), \quad x(0) \in \mathbb{R}^n_{\geq 0}, \tag{4}$$

where x represents the concentration vector of the species, and the monomial vector mapping $\psi^Y(x)$ is defined as

$$\psi_j^Y(x) = \prod_{i=1}^{n} x_i^{[Y]_{ij}} \quad j = 1, \ldots, m. \tag{5}$$

At this point, we can make a connection between a nonnegative polynomial ODE and the dynamics of CRNs as follows: a nonnegative polynomial ODE in (1) can be transformed into the form of (4), i.e.

$$M\varphi(x) = Y A_\kappa \psi^Y(x), \tag{6}$$

if and only if the following condition is fulfilled

$$\begin{aligned} &\text{if } [M]_{ij} \leq 0, \text{then } [B]_{ij} > 0, \\ &i = 1, \ldots, n, \quad j = 1, \ldots, m. \end{aligned} \tag{7}$$

Example: This small example illustrates the CRN definitions. Consider the following CRN:

$$3\,\mathrm{X}_2 \xrightarrow{\mathrm{k}_{12}} 3\,\mathrm{X}_1 \xrightarrow{\mathrm{k}_{23}} 2\,\mathrm{X}_1 + \mathrm{X}_2.$$

Then, we have two species (X_1 and X_2) and three complexes ($3X_2$, $3X_1$ and $2X_1 + X_2$), based on these the CRN can be encoded as

$$Y = \begin{matrix} \\ x_1 \\ x_2 \end{matrix} \begin{matrix} \begin{matrix} C_1 & C_2 & C_3 \end{matrix} \\ \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 1 \end{bmatrix} \end{matrix}, \quad A_\kappa = \begin{matrix} \\ C_1 \\ C_2 \\ C_3 \end{matrix} \begin{matrix} \begin{matrix} C_1 & C_2 & C_3 \end{matrix} \\ \begin{bmatrix} -k_{12} & 0 & 0 \\ k_{12} & -k_{23} & 0 \\ 0 & k_{23} & 0 \end{bmatrix} \end{matrix},$$

and finally, by setting $k_{12} = 1$ and $k_{23} = 3$, we have

$$M = \begin{bmatrix} 3 & -3 & 0 \\ -3 & 3 & 0 \end{bmatrix}, \quad \psi_j^Y(x) = \begin{bmatrix} x_2^3 \\ x_1^3 \\ x_2^2 x_2 \end{bmatrix}. \tag{8}$$

Although it is not discussed in this paper, when the underlying reaction graphs are also computed, then one must take into account that multiple reaction graphs may exist for the same kinetic system. The details of how this nonunique relationship, between the dynamics and the reaction graphs, hinders the system identification process can be found in [3], [8], [9].

It should be emphasised that a kinetic system with mass action kinetics is always linear in parameters. Therefore, a linear regression problem can be formulated to estimate its parameters. However, depending on the amount of data available, we encounter an overdetermined or underdetermined linear regression problem, as described in the next section.

III. Automated Model Building

Our goal is to estimate the parameter of a system described in (1), namely the entries of M. We then need to distinguish between two cases:

1) The entries of $\varphi(x)$ are known, thus we know exactly which monomials are needed to describe the dynamics. In that case, we only need to estimate which monomials are participating in a given state equation. More importantly, we need to correctly estimate those that are not part of a given state equation, e.g. estimating the position of the zeros in matrix M.
2) In a more challenging scenario, the $\varphi(x)$ is not known *a priori*, but only the participating chemical species: X_i, $i = 1, \ldots, n$. In such case, we assemble $\varphi(x)$ from all possible monomials up to a certain number of species, e.g. the set of all possible pairs of the species:

   $$\{X_i + X_j \mid i = 1, \ldots, n, \quad j = 1, \ldots, n\}.$$

   In this case, excluding all the monomials that are not participating in the governing equations is the most important task.

In both cases, the entries of $\varphi(x)$ that are encoding the number of possible monomials may exceed the number of measurements, i.e. we have more parameters than measured data points.

Solving the problem in the first case is common practice with the assumption that more measurements than parameters are available. Usually, an ODE model is formulated, since the model structure is known, and then it is simulated with parameter guesses generated by the parameter estimation algorithm. The speed of this approach is heavily dependent on two factors: the speed of the ODE solvers (some parameter combinations can create very stiff ODEs, which slows down the numerical integration), and the convergence speed of the optimisation solver (usually gradient information is used, which is generated by numerical approximation or by sensitive equations, and either can slow down convergence).

Solving the problem in the second case is usually avoided because of data scarcity or computational challenges.

In this paper, we show a solution for both estimation problems, by focusing on the second case; the first case naturally falls into the developed solution. Moreover, we assume that we have an underdetermined problem. i.e. infinitely many of solutions exist. To tackle this, we need to impose further constraints on the underdetermined problem to find an acceptable solution.

A. Parameter estimation model

Our goal is to estimate the parameter of the following system

$$y = Aw + \nu, \tag{9}$$

where $y \in \mathbb{R}^p$ is the measurement vector, the matrix $A \in \mathbb{R}^{p\times m}$ is the regressor, the parameter vector is $w \in \mathbb{R}^m$ and $\nu \sim \mathcal{N}(0, \lambda I)$ is the measurement noise.

Since the kinetic system in (1) is linear in parameters, we can formulate a linear regression for each row of the matrix M independently. For the derivation, we will focus on just one state variable, and at the end of this section, we will estimate all rows of the matrix M at once.

Each state can be written as

$$\dot{x}_i = [M]_{i,\cdot}\varphi(x), \quad i = 1, \dots, n, \tag{10}$$

and we have a measurement vector of each state such that

$$X = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_p) & x_2(t_p) & \dots & x_n(t_p) \end{bmatrix}, \tag{11}$$

where, t_k is the time of the measurement at sampling time point k.

Additionally, we compute the time derivate of x as

$$y_i = \frac{d}{dt}x_i, \quad i = 1, \dots, n. \tag{12}$$

Then, we generate the measurement data as

$$y^i = \begin{bmatrix} y_i(t_2) & y_i(t_3) & \dots & y_i(t_p) \end{bmatrix}^\top, \quad i = 1, \dots, n, \tag{13}$$

where superscript i denotes the ith state.

The mapping $\varphi(x)$ is evaluated on the measurement matrix $X \in \mathbb{R}^{p\times m}$, namely

$$[A]_{i,j} = \varphi_j([X]_{i,\cdot}), \quad i = 1, \dots, p, \quad j = 1, \dots, m. \tag{14}$$

With this framework, we can formulate the linear regression problem as

$$\hat{w}^i = \underset{w}{\operatorname{argmin}} ||y^i - Aw^i||_2, \quad i = 1, \dots, n. \tag{15}$$

Once we have a solution for each state equation, we have

$$\hat{M}_{i,\cdot} = \hat{w}^{i\top}, \quad i = 1, \dots, n. \tag{16}$$

The zero entries in $\hat{w}^i$ control which monomial in $\varphi(x)$ gets selected into the ith state equation.

B. Sparse Bayesian Learning

Since we assume that we have more parameters than measurements ($p < m$), this linear regression problem is underdetermined.

To solve this problem, we use the so-called Sparse Bayesian Learning (SBL) method, which was developed in [10], [11]. The underlying algorithm is outlined here for reference, but the details can be found in [10]. The SBL algorithm has been used for parameter estimation and model building in [12], [13], [14]

Until the end of this section, we drop the i superscript for clarity.

The estimation of w for each state can be formulated in a Bayesian setting, where we place a prior distribution on the parameters as

$$p(w) = \prod_{j=1}^{m} p(w_j), \quad p(w_j) = \max_{\gamma_j \geq 0} \mathcal{N}(0, \gamma_j)\zeta(\gamma_j), \tag{17}$$

where $\zeta(\gamma_j)$ is a nonnegative function.

The likelihood function is

$$p(y|w) \propto \exp\left(-\frac{1}{2}||y - Aw||_2^2\right). \tag{18}$$

The goal of the algorithm below is to compute the value of γ_j for each w_j, while trying to maximise the number of γ_j with zero value.

To do so the following algorithm was developed in [10]:

- Step 1: initialise each $z_j = 1, j = 1, \dots, m$
- Step 2: $\hat{w} = \operatorname{argmin}_w ||y - Aw||_2^2 + 2\lambda \sum_j z_j^{-1/2}|w_j|$
- Step 3: compute $\gamma_j^{opt} = z_j^{-1/2}|\hat{w}_j|, j = 1, \dots, m$
- Step 4: compute $z^{opt} = \nabla_\gamma \log|\Sigma_y|$
- Step 5: iterate Step 2, 3 and 4 until γ converges to some value.

In this algorithm, λ is a regularisation term, which is determined by the variance of the measurement noise (assumed to be known *a priori*), and $\sum_y = \lambda I + A\Gamma A^\top$, where $\Gamma = \text{diag}[\gamma]$.

Step 2 is a convex optimisation problem, see [10]. Interestingly, in the first iteration, the algorithm is a LASSO problem [15], which is then improved at each iteration of the algorithm.

C. Computing the matrix M at once

Most of the system properties depend on M and therefore, our goal is to impose constraints on matrix M. To achieve that we need to compute all entries of M in one optimisation problem.

Consider the following matrix regression problem

$$Y = AW + \mathcal{V}, \tag{19}$$

where

$$Y = \begin{bmatrix} y^1 & y^2 & \dots & y^n \end{bmatrix}, \tag{20}$$
$$W = \begin{bmatrix} w^1 & w^2 & \dots & w^n \end{bmatrix}, \tag{21}$$

matrix A is the same as before and $\nu^i \sim \mathcal{N}(0, \lambda_i I)$.

The cost function in Step 2 above can be rewritten as

$$\hat{W} = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n} \left(||y^i - Aw^i||_2^2 + 2\lambda_i \sum_{j=1}^{m} (z_j^i)^{-1/2} |w_j^i| \right). \tag{22}$$

Because we sum up convex functions the new cost function is also convex.

Using the Sparse Bayesian Learning and the underlying optimisation problem (22) extended to the multi-state case, we can automatically build models from time-series data. However, this approach has two caveats:

- Only linear parameters can be estimated. For example, estimating the cooperativity of a Hill function is not possible with the current setup. Some solutions to this problem have been proposed in [16].
- All state variables must be measured. This is a severe limitation to our approach, however, designing observers for the unmeasured states could alleviate this problem.

We are now ready to impose constraints on $\hat{W}$.

IV. ENFORCING SYSTEM PROPERTIES

The SBL framework solves an underdetermined system of equations which has infinitely many solutions. Among those, the algorithm in Section III-C aims to find the sparsest solution. Even if the sparest solution fits the data, it might not represent a nonnegative system or a kinetic system. Moreover, the resulting ODE might be unstable. To tackle these problems, we develop constraints that encode certain system properties and are then used to define an optimisation problem under such constraints. In particular, we investigate here how to use constraints to enforce a solution, which represents a kinetic system, and how to find a system which has bounded trajectories.

A. Enforcing kinetic systems

The SBL framework builds first-order ODEs, but we want to restrict the results to kinetic systems, which are nonnegative and have a CRN representation [8]. In Section II, we have introduced the definition of kinetic systems using CRNs. In this section, we start from a polynomial ODE and state under which conditions a CRN representation, e.g. a directed graph, exists. This is called kinetic realisability of nonnegative systems or inverse problem of reaction kinetics [2], [17].

Consider the following ODE as

$$\dot{x} = f(x), \quad x(0) \in \mathbb{R}^n_{\geq 0}, \tag{23}$$

then, a necessary and sufficient condition for kinetic realisability of a polynomial ODE is the following:

$$f_i(x) = -x_i g_i(x) + h_i(x), \qquad i = 1, \dots .n, \tag{24}$$

where g_i and h_i are polynomials with nonnegative coefficients.

The criteria in (24) was reported in [17], along with an algorithm to construct a CRN realisation from (24).

The structure of (24) enables us to formulate a convex constraint for the optimisation problem in (22).

The constraint for the ith state equation is the following

$$\text{if } x_i \notin \varphi_j(x), \text{ then } \; w_j^i(x) \geq 0, \quad j = 1, \dots, m. \tag{25}$$

Such set of inequality constraints are added to the optimisation problem for each state variables.

B. Enforcing bounded trajectories

Due to the structure of right-hand side in (1), we can construct a Lyapunov-type function (see e.g. [18], [19]) for a kinetic system, which then guaranties that the system has bounded trajectories.

Let us define

$$V(x) = k^\top x \tag{26}$$

where $k \in \mathbb{R}^n$ is a column vector and $k > 0$ (element-wise) and x is the solution of (1).

Then, the time derivate of V is

$$\dot{V}(x) = k^\top \dot{x} = k^\top M \varphi(x). \tag{27}$$

To have bounded trajectories, $\dot{V}$ needs to satisfy

$$\dot{V} = k^\top M \varphi(x) \leq 0 \tag{28}$$

Since all the parameters of a kinetic system are in M, we can formulate the following bilinear constraint, assuming that M is known and k is a decision variable,

$$\begin{aligned} k^\top M &\leq 0 \\ k &> 0 \end{aligned} \tag{29}$$

where $\leq$ and $>$ are element-wise inequalities.

If such a $k > 0$ exists for a kinetic system, then the underlying CRN is called subconservative [20], and the kinetic system has bounded trajectories. Further system properties of such systems are explored in [3], [8].

Example: Let us use the example form Section II, where

$$M = \begin{bmatrix} 3 & -3 & 0 \\ -3 & 3 & 0 \end{bmatrix},$$

then finding $k > 0$ is easy for this system, e.g.

$$k = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

would statisfy (29).

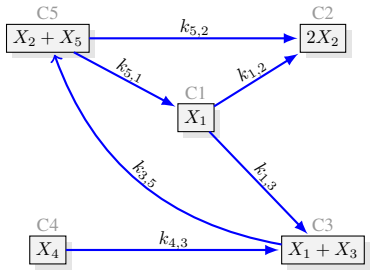

Fig. 1. This example network is taken from [21]. The parameters of the network are $k_{12} = 0.3386$, $k_{13} = 0.8244$, $k_{51} = 0.8496$, $k_{52} = 0.4290$, $k_{35} = 0.7364$ and $k_{43} = 0.5630$.

Incorporating in the SBL framework: Within the SBL framework, the parameters of the kinetic system are not known, thus M is unknown. In such a case, both k and W are decision variables, thus the optimisation problem with cost function in (22) and constrains in (29) is not convex, and we need to resort to general solvers, e.g. `fmincon` in Matlab.

V. EXAMPLES

The example used in this section was presented in [21] as a benchmark problem for network inference. The network structure is shown in Figure 1 and its CRN formulation is given as

$$Y = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

and

$$A_\kappa = \begin{bmatrix} -1.163 & 0 & 0 & 0 & 0.8492 \\ 0.3386 & 0 & 0 & 0 & 0.4290 \\ 0.8244 & 0 & -0.7364 & 0.5631 & 0 \\ 0 & 0 & 0 & -0.5631 & 0 \\ 0 & 0 & 0.7364 & 0 & -1.2782 \end{bmatrix}. \tag{31}$$

This CRN has 5 complexes and 6 reactions and the parameters of the orignal system are

$$M = \tag{32}$$
$$\begin{array}{c} \begin{matrix} x_1 & x_2x_2 & x_1x_3 & x_4 & x_2x_5 \end{matrix} \\ \begin{bmatrix} -0.3386 & 0 & -0.7364 & 0.5630 & 0.8496 \\ 0.6772 & 0 & 0.7364 & 0 & -0.4206 \\ 0.8244 & 0 & -0.7364 & 0.5630 & 0 \\ 0 & 0 & 0 & -0.5630 & 0 \\ 0 & 0 & 0.7364 & 0 & -1.2786 \end{bmatrix} \end{array}.$$

A. Estimating a kinetic system

Using the parameters in (32), we have generated five different set of initial conditions (uniformly sampled between 0 and 50) and simulated the system for $T = 0.5$ min. The time derivative of measured data was also calculated, then zero mean additive Gaussian noise with $\sigma^2 = 0.1$ was added to y. Using Latin-hypercube sampling, we have selected a total of $p = 45$ measurements for each state variables.

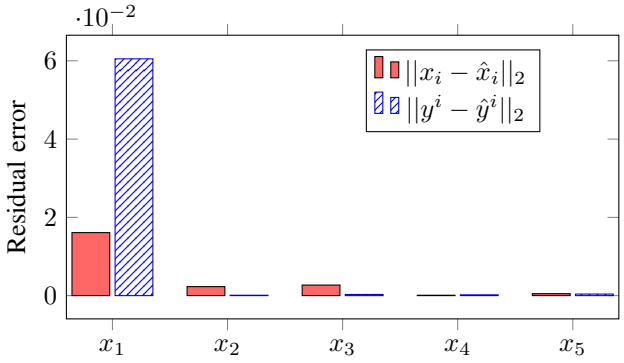

Fig. 2. Trajectory differences between the original and reconstructed model in (33) (red), as well as the residual error on the data fit (blue) for each states. The error metrics were calculated using five randomly generated initial conditions and both models were simulated for 1 min.

This example has five species and the dictionary contains all possible pairs and triplets of these species, plus a constant term, thus the dictionary consists of $m = 56$ functions.

We evaluated these dictionary functions on the generated measured data using (14).

At this point, we have $y^i \in \mathbb{R}^p$, $i = 1, \ldots, n$ and $A \in \mathbb{R}^{p \times m}$. Based on this, we can then solve the optimisation problem with the cost function (22) and constraint (25) using the Mosek solver. As the result we have

$$\hat{M} = \tag{33}$$
$$\begin{array}{c} \begin{matrix} x_1 & x_1x_3 & x_4 & x_2x_5 & x_5 \end{matrix} \\ \begin{bmatrix} 0 & -0.73804 & 0 & 0.84692 & 0.16033 \\ 0.6772 & 0.7364 & 0 & -0.4206 & 0 \\ 0.8244 & -0.7364 & 0.563 & 0 & 0 \\ 0 & 0 & -0.563 & 0 & 0 \\ 0 & 0.7364 & 0 & -1.2786 & 0 \end{bmatrix} \end{array}.$$

Figure 2 shows the residual error in the data fit (blue) and the trajectory differences between the original and the reconstructed system (red) for each states. With this current setup, we can restrict the reconstructed systems to the nonnegative orthant.

B. Estimating a kinetic system with bounded trajectories

In this second example, we want to estimate M, but with constraint (29) in order to have bounded the state trajectories, however this makes the optimisation problem non-convex. For this a proof-of-concept example, we used the following settings.

The dictionary contains all possible pairs of the species and a constant term, which gives $m = 21$. We generate more data using the previous settings, which gives us $p = 200$. Using `fmincon` in Matlab, we estimate the matrix M in (35) and

$$k = \begin{bmatrix} 0.1037 & 0.0033 & 0.0330 & 0.2459 & 0.0811 \end{bmatrix}^\top. \tag{34}$$

The reconstructed model has three extra terms compared to the original model, but all state variables are nonnegative and the state trajectories are bounded. Figure 3 shows the residual

$$
\begin{array}{c}
\begin{array}{ccccccccccccccccccccc}
1 & x_1 & x_2 & x_3 & x_4 & x_5 & x_1x_1 & x_1x_2 & x_1x_3 & x_1x_4 & x_1x_5 & x_2x_2 & x_2x_3 & x_2x_4 & x_2x_5 & x_3x_3 & x_3x_4 & x_3x_5 & x_4x_4 & x_4x_5 & x_5x_5
\end{array} \\
\left[\begin{array}{ccccccccccccccccccccc}
0 & -0.259 & 0 & 0 & 0.401 & 0 & 0 & 0 & -0.741 & 0 & 0 & 0 & 0 & 0 & 0.848 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.648 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & 0 & 0 & -0.433 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.647 & 0 & 0 & 0.521 & 0.089 & 0 & 0 & -0.656 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.144 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.440 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.088 & 0 & 0 & 0.716 & 0 & 0 & 0 & 0 & 0 & -1.277 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array}
\tag{35}
$$

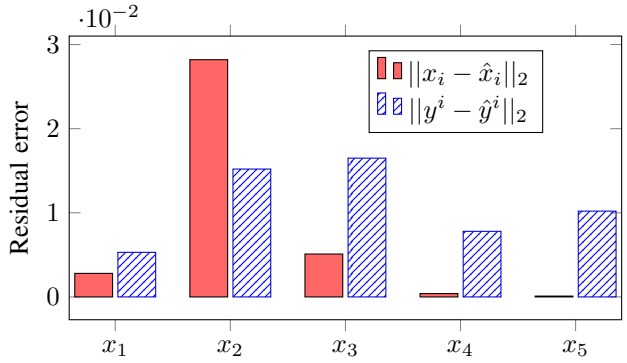

Fig. 3. Trajectory differences between the original and reconstructed model in (35) (red), as well as the residual error on the data fit (blue) for each states. The error metrics were calculated using five randomly generated initial conditions and both models were simulated for 1min.

error in the data fit (blue) and the trajectory differences between the original and the reconstructed system (red).

VI. CONCLUSIONS

In this paper, we have demonstrated an extension to the Sparse Bayesian Learning framework where the parameters of all state equations are estimated simultaneously. Additionally, we have reported constraints that guide the automatic model building in order to ensure that the resulting models have certain system properties. In particular, we have shown how to ensure nonnegative state variables or bounded state trajectories in the reconstructed model. As part of ongoing work we are investigating similar constraints to ensure stability or structural identifiability.

ACKNOWLEDGMENT

This work was developed as part of the project called COSY- BIO (Control Engineering of Biological Systems for Reliable Synthetic Biology): H2020-FETOPEN-2016-2017, project number 766840. This work was also supported by the U.K. EPSRC Fellowship EP/M002187/1 to Dr Stan. The authors thank Petra Rovo for her thoughtful comments.

REFERENCES

[1] L. Bandiera, Z. Hou, V. Kothamachu, E. Balsa-Canto, P. Swain, and F. Menolascina, "On-line optimal input design increases the efficiency and accuracy of the modelling of an inducible synthetic promoter," *Processes*, vol. 6, no. 9, p. 148, sep 2018.

[2] W. M. Haddad, V. Chellaboina, and Q. Hui, *Nonnegative and Compartmental Dynamical Systems*. Princeton University Press, 2010.

[3] B. Ács, G. Szlobodnyik, and G. Szederkényi, "A computational approach to the structural analysis of uncertain kinetic systems," *Computer Physics Communications*, vol. 228, pp. 83–95, 2018.

[4] D. D. Vecchio and R. M. Murray, *Biomolecular Feedback Systems*. Princton University Press, 2014.

[5] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[6] A. Raue, M. Schilling, J. Bachmann, A. Matteson, M. Schelke, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, U. Klingmüller, and J. Timmer, "Lessons learned from quantitative dynamical modeling in systems biology," *PLoS ONE*, vol. 8, no. 9, p. e74335, sep 2013.

[7] A. Gábor, K. M. Hangos, and G. Szederkényi, "Linear conjugacy in biochemical reaction networks with rational reaction rates," *Journal of Mathematical Chemistry*, vol. 54, no. 8, pp. 1658–1676, may 2016.

[8] B. Ács, G. Szederkényi, Z. Tuza, and Z. A. Tuza, "Computing all possible graph structures describing linearly conjugate realizations of kinetic systems," *Computer Physics Communications*, vol. 204, pp. 11–20, 2016.

[9] G. Szederkényi, J. R. Banga, and A. A. Alonso, "Inference of complex biological networks: distinguishability issues and optimization-based solutions," *BMC Systems Biology*, vol. 5, p. 177, 2011.

[10] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in neural information processing systems*, 2008, pp. 1625–1632.

[11] D. Wipf and S. Nagarajan, "Iterative reweighted l1 and l2 methods for finding sparse solutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 317–329, 2010.

[12] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, mar 2016.

[13] W. Pan, "Bayesian learning for nonlinear system identification," Ph.D. dissertation, Imperial College London, 2017.

[14] Z. A. Tuza and G.-B. Stan, "Characterization of biologically relevant network structures form time series data," in *57th IEEE Conference on Decision and Control, Miami Beach, FL, USA, December 17-19, 2018*, 2018.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[16] Q. Zhu, "An implicit least squares algorithm for nonlinear rational model parameter estimation," *Applied Mathematical Modelling*, vol. 29, no. 7, pp. 673–689, jul 2005.

[17] V. Hárs and J. Tóth, "On the inverse problem of reaction kinetics," in *Qualitative Theory of Differential Equations*, ser. Coll. Math. Soc. J. Bolyai, M. Farkas and L. Hatvani, Eds. North-Holland, Amsterdam, 1981, vol. 30, pp. 363–379.

[18] P. Polcz, T. Peni, and G. Szederkenyi, "Reduced linear fractional representation of nonlinear systems for stability analysis," *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 37–42, 2018.

[19] M. A. Al-Radhawi and D. Angeli, "Construction of robust lyapunov functions for reaction networks," in *2016 European Control Conference (ECC)*. IEEE, jun 2016.

[20] M. D. Johnston, D. F. Anderson, G. Craciun, and R. Brijder, "Conditions for extinction events in chemical reaction networks with discrete state spaces," *Journal of Mathematical Biology*, vol. 76, no. 6, pp. 1535–1558, sep 2017.

[21] E. August and A. Papachristodoulou, "Efficient, sparse biological network determination," *BMC Systems Biology*, vol. 3, p. 25, 2009.