Reconstruction of Arbitrary Biochemical Reaction Networks: A Compressive Sensing Approach

Wei Pan, Ye Yuan[†] and Guy-Bart Stan

Abstract

Reconstruction of biochemical reaction networks is a central topic in systems biology which raises crucial theoretical challenges in system identification. Nonlinear Ordinary Differential Equations (ODEs) that involve polynomial and rational functions are typically used to model biochemical reaction networks. Such nonlinear models make the problem of determining the connectivity of biochemical networks from time-series experimental data quite difficult. In this paper, we present a network reconstruction algorithm that can deal with model descriptions under the form of polynomial and rational functions. Rather than identifying the parameters of linear or nonlinear ODEs characterised by pre-defined equation structures, our methodology allows us to determine the nonlinear ODEs structure together with their associated reaction constants. To solve the network reconstruction problem, we cast it as a Compressive Sensing (CS) problem and use Bayesian Sparse Learning (BSL) algorithms as an efficient way to obtain its solution.

I. INTRODUCTION

A long standing problem in systems biology is to reconstruct biochemical reaction networks. Reconstruction means to identify both the topology and the parameter values of some biochemical reaction networks. More specifically, network reconstruction tries to recover the set of nonlinear ODEs associated with the biochemical processes from time-series experimental data. A naive reconstruction method consists in searching among all possible reactions the few that seem consistent with the time series. The associated computational burden of such a naive approach is typically horrendous even for network of modest dimensions. Within the systems and control community, identification of biochemical reaction networks, genetic regulatory networks in particular, are quite active research areas [1], [17], [18], [20], [22], [29].

Many linear and nonlinear functions can be used to describe the dynamics of biochemical reaction networks in terms of biochemical kinetic laws, e.g., as first-order functions $f([S]) = [S]$, mass action functions $f([S_1], [S_2]) = [S_1] \cdot [S_2]$, Michaelis-Menten functions $f([S]) = V_{\max}[S]/(K_M + [S])$, Hill functions $f([S]) = V_{\max}[S]^n/(K_M^n + [S]^n)$. Furthermore, it is not uncommon that a single gene is regulated by more than one transcription factor. In such situations, the combined effect of these regulators on gene expression needs to be described by a multi-dimensional input function. Such input functions typically take the form of ratio of polynomials involving the concentrations of the input transcription factors x_i, $i = 1, \dots, n$, for example, $f(x_1, \dots, x_n) = \sum_i \beta_i (x_i/K_i)^{n_i} / (1 + \sum_i \beta_i (x_i/K_i)^{m_i})$, where K_i is the activation or repression coefficient for the transcription factor x_i, β_i is its maximal contribution to gene expression, and the Hill coefficients are $n_i = m_i > 0$ for activators and $n_i = 0$, $m_i > 0$ for repressors. These

W. Pan, Y. Yuan and G.-B. Stan are with Centre for Synthetic Biology and Innovation and the Department of Bioengineering, Imperial College London, United Kingdom.

Y. Yuan is also with Control Group, Department of Engineering, University of Cambridge, United Kingdom.

[†]For correspondence, yy311@cam.ac.uk.

types of functions have been shown to appropriately describe experimentally determined input functions [21]. More complicated expressions are also possible, e.g., if the different transcription factors interact with each other at the protein level [4].

Our main objective is, given experimental time-series data, to identify both the interconnection topology (the form of the nonlinear functions) and their associated parameters. Without prior knowledge on model structure, this is still an open problem in system identification.

During his plenary talk at the 50th IEEE CDC, Prof. Lenart Ljung emphasised on four opportunities for further research and development in system identification. Two of these were: (a) the use of sparse datasets and (b) the use of machine learning approaches [15]. The approach that we consider in this paper is aligned with these two recommendations. Specifically, our approach draws inspiration from the fields of signal processing and machine learning, by combining compressive sensing and Bayesian sparse learning to offer an efficient method for network reconstruction.

The paper is organised as follows. In Section II, we introduce the type of biochemical reaction networks model we consider in this paper, i.e., discrete Langevin ODEs. In Section III, we formulate the network reconstruction problem associated with the model class proposed in Section II. In Section IV, we show how the reconstruction problem can be converted into a compressive sensing problem. In Section V, we show how Bayesian sparse learning algorithms can be used to solve the compressive sensing problem. In Section VI, we apply our method to the reconstruction of the repressilator and show how it can be reconstructed almost exactly using the compressive sensing framework. Finally, in Section VII, we conclude and discuss several future problems that we plan to address.

II. MODEL FORMULATION

We consider dynamical systems of the following general form:

$$\frac{dx}{dt} = Sf(x), \tag{1}$$

where $x \in \mathbb{R}^n$, $S \in \mathbb{R}^{n \times L}$, and $f(\cdot) \in \mathbb{R}^L$ is a vector of L unknown functions. This set of functions can, for example, include mass action kinetic terms under the form of product of monomials, monotonically increasing or monotonically decreasing Hill functions, simple linear terms, etc. Let neither the value of the entries nor the structure of matrix S be known. Given experimental data in terms of time series of state x, our objective is to identify the values in S.

Applying standard Euler discretisation to (1), we obtain:

$$x(t_{k+1}) = x(t_k) + (t_{k+1} - t_k)Sf(x(t_k)), \tag{2}$$

Assuming that the discretisation step $t_{k+1} - t_k = \epsilon$ is constant for all k, and defining the error between two successive state values as $e(t_{k+1}) \triangleq x(t_{k+1}) - x(t_k)$, we can write (2) under the general form

$$e(t_{k+1}) = \mathbf{v}^T f(x(t_k)), \tag{3}$$

where $e = [e_1, e_2, \ldots, e_n]^T \in \mathbb{R}^{n \times 1}$ denotes the error vector; $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n \times 1}$ denotes the state vector; $\mathbf{v}^T = \begin{bmatrix} v_{11} & \ldots & v_{L1} \\ \vdots & \vdots & \vdots \\ v_{1n} & \ldots & v_{Ln} \end{bmatrix} = \epsilon S \in \mathbb{R}^{n \times L}$; and $f(x) = [f_1(x), f_2(x), \ldots, f_L(x)]^T \in \mathbb{R}^{L \times 1}$ is the vector field.

Typically biochemical reaction networks exhibit stochastic behaviours due to intrinsic and extrinsic noise at the gene expression level. Furthermore, the acquisition of data is typically affected by the inaccuracy of the measurement devices as well as by unpredictable environmental perturbations. It is therefore more realistic to consider *Langevin nonlinear dynamics*:

$$e(t_{k+1}) = \mathbf{v}^T f(x(t_k)) + \xi(t_k). \tag{4}$$

$\xi(t_k) \in \mathbb{R}^n$ represents energy-bounded process noises which are assumed to be independent, to be distributed according to normal probability distributions:

$$\begin{aligned} E[\xi(t_k)] &= 0, \\ E[\xi(t_k)\xi^T(t_j)] &= Q_k \delta_{kj}, \end{aligned}$$

where

$$\delta_{kj} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases} \tag{5}$$

In the following section, we shall propose an algorithm that uses time-series data to reconstruct the eq. (1).

III. RECONSTRUCTION PROBLEM FORMULATION

A. Problem Formulation

Consider the discrete-time model in eq. (4):

$$e^T(t_{k+1}) = f^T(x(t_k))\mathbf{v} + \xi^T(t_k), \tag{6}$$

Assuming that M successive data samples are obtained from (6) and defining

$$\begin{aligned}
\mathbf{y} &\triangleq \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} e(t_1) \dots & e(t_M) \end{bmatrix}^T \\
&= \begin{bmatrix} e_1(t_1) & \dots & e_n(t_1) \\ e_1(t_2) & \dots & e_n(t_2) \\ \vdots & \ddots & \vdots \\ e_1(t_M) & \dots & e_n(t_M) \end{bmatrix} \in \mathbb{R}^{M \times n}, \\
\Theta &\triangleq \begin{bmatrix} f_1(x(t_0)) & \dots & f_L(x(t_0)) \\ f_1(x(t_1)) & \dots & f_L(x(t_1)) \\ \vdots & \ddots & \vdots \\ f_1(x(t_{M-1})) & \dots & f_L(x(t_{M-1})) \end{bmatrix} \\
&= \begin{bmatrix} f^T(x(t_0)) \\ f^T(x(t_1)) \\ \vdots \\ f^T(x(t_{M-1})) \end{bmatrix} \in \mathbb{R}^{M \times L}, \\
\mathbf{v} &\triangleq \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix} \in \mathbb{R}^{L \times n}, \\
\Xi &\triangleq \begin{bmatrix} \xi_1 & \xi_2 & \dots & \xi_n \end{bmatrix} \\
&= \begin{bmatrix} \xi_1(t_0) & \dots & \xi_n(t_0) \\ \xi_1(t_1) & \dots & \xi_n(t_1) \\ \vdots & \ddots & \vdots \\ \xi_1(t_{M-1}) & \dots & \xi_n(t_{M-1}) \end{bmatrix} \in \mathbb{R}^{M \times n}.
\end{aligned}$$

Eq. (6) can be written

$$\mathbf{y} = \Theta\mathbf{v} + \Xi, \tag{7}$$

or

$$\mathbf{y}_i = \Theta\mathbf{v}_i + \xi_i,\ (i = 1, \dots, n). \tag{8}$$

We want to find $\mathbf{v}_i$ given measured data stored in $\mathbf{y}_i$. This a typical linear regression problem that can be solved using standard least square approaches, provided that the structure of the nonlinearities in the model are known, i.e., provided that Θ is known. However, in most cases, these nonlinearities are unknown or difficult to assume *a priori*, and as a consequence, Θ itself is unknown and therefore $\mathbf{v}_i$ cannot be solved from eq. (8).

Depending on the field for which the dynamical model needs to be built, only a few typical nonlinearities specific to this field need to be considered. For example, the class of models that arise from biochemical reaction networks typically involve nonlinearities that capture fundamental biochemical kinetic laws, e.g., first-order degradation functions, mass-action kinetics, Hill and Michaelis-Menten functions. In what follows we gather in a matrix Φ similar to Θ the set of *all* candidate basis functions that we want to consider for reconstruction:

$$\Phi \triangleq \begin{bmatrix} F_1(x(t_0)) & \dots & F_N(x(t_0)) \\ F_1(x(t_1)) & \dots & F_N(x(t_1)) \\ \vdots & \ddots & \vdots \\ F_1(x(t_{M-1})) & \dots & F_N(x(t_{M-1})) \end{bmatrix} \in \mathbb{R}^{M\times N} \tag{9}$$

This leads to equations similar to (8):

$$\mathbf{y}_i = \Phi\mathbf{w}_i + \xi_i,\ (i = 1, \dots, n). \tag{10}$$

where

$$\begin{aligned} \mathbf{w} &= \begin{bmatrix} \mathbf{w}_1 & \dots & \mathbf{w}_n \end{bmatrix} \qquad (11) \\ &= \begin{bmatrix} \omega_{11} & \dots & \omega_{1n} \\ \vdots & \vdots & \vdots \\ \omega_{N1} & \dots & \omega_{Nn} \end{bmatrix} \in \mathbb{R}^{N\times n}. \end{aligned}$$

We will introduce in section IV a method that allows us to reconstruct $\mathbf{w}_i$ from time-series observations of x and $\mathbf{y}_i$. Before explaining this method, we introduce in the following two sections the most commonly used nonlinear functions that appear in nonlinear ODEs used to model biochemical reaction networks and explain how, based on these functions, the reconstruction problem amounts to infer Φ.

B. Polynomial Input Functions

A *Monomial* m_α in n variables is a function defined as $m_\alpha(x) = x^\alpha \triangleq x_1^{\alpha_1}x_2^{\alpha_2}\dots x_n^{\alpha_n}$ for $\alpha \in \mathbb{Z}_+^n$. The degree of a monomial is defined as $\deg m_\alpha := \sum_{i=1}^n \alpha_i$. Polynomials being decomposable into sums of monomial terms, the elements $F_j(x)$ appearing in the basis function matrix Φ can be represented as monomials of the form:

$$F_j(x) = x_1^{\alpha_1}x_2^{\alpha_2}\dots x_n^{\alpha_n}, \tag{12}$$

where x_k $(k = 1, \dots, n)$ is the kth component of the dynamical variable x. As an illustration, we consider the case of $n = 2$ variables x_1, x_2. In that case, we have the following basis functions to represent polynomials: $F_1(x) = x_1^0 x_2^0 = 1,\ F_2(x) = x_1^1 x_2^0 = x_1,\ F_3(x) = x_1^0 x_2^1 = x_2,\ \ldots,\ F_N(x) = x_1^{s_1} x_2^{s_2}$, where the exponents s_1 and s_2 are the maximal possible stoichiometric coefficients for x_1 and x_2 in biochemical reactions [19].

Given an upper bound on the degree of the polynomials that we want to consider as potential candidates in the reconstruction process, we thus need to consider as elements $F_j(x)$ of Φ all possible independent monomials of degree less or equal to the maximal degree of the candidate polynomial.

C. Rational Input Functions

Another common type of nonlinearities in dynamical models of biochemical reaction networks are the activation and repression Hill and Michaelis-Menten functions. To include these nonlinearities as potential candidates in the reconstruction process, we consider the general multi-dimensional input function introduced in Section I

$$G_j(x) = \frac{\sum_j \beta_j (x_j/K_j)^{n_j}}{1 + \sum_j \beta_j (x_j/K_j)^{m_j}},$$

where β_j and K_j are parameters to be identified and the exponents n_j and m_j are assumed to be fixed. The cases of gene activation or repression are encompassed by this general form: gene activation corresponds to the situation where $n_j = m_j > 0$ whereas gene repression corresponds to the case where $n_j = 0$.

To give an idea on how to deal with rational rate terms, we assume a general model of the form as described in (3):

$$e_i(t_{k+1}) = \frac{\sum_j \alpha_{ij} (x_{ij}(t_k)/K_{ij})^{n_{ij}}}{1 + \sum_j \beta_{ij} (x_{ij}(t_k)/K_{ij})^{m_{ij}}} - \gamma_i x_i(t_k) + C_i. \tag{13}$$

By multiplying both sides of equation (13) by $1 + \sum_j \beta_{ij} (x_{ij}(t_k)/K_{ij})^{m_{ij}}$, we obtain

$$\begin{aligned} & \left[1 + \sum\nolimits_j \beta_{ij} (x_j(t_k)/K_{ij})^{m_{ij}}\right] e_i(t_{k+1}) \\ = \ & -\gamma_i \left[1 + \sum\nolimits_j \beta_{ij} (x_j(t_k)/K_{ij})^{m_{ij}}\right] x_i(t_k) \\ & + \left[1 + \sum\nolimits_j \beta_{ij} (x_j(t_k)/K_{ij})^{m_{ij}}\right] C_i \\ & + \sum\nolimits_j \alpha_{ij} (x_j(t_k)/K_{ij})^{n_{ij}} \end{aligned}$$

1) Activation Hill Functions: $n_{ij} = m_{ij}$: We let

$$\begin{aligned} & \hat{\gamma}_i = -\gamma_i, \quad \bar{\alpha}_{ij} = \alpha_{ij}/K_{ij}^{n_{ij}}, \quad \bar{\beta}_{ij} = \beta_{ij}/K_{ij}^{n_{ij}}. \\ & \hat{\alpha}_{ij} = \bar{\alpha}_{ij} + C_i \bar{\beta}_{ij}, \quad \hat{\beta}_{ij} = \hat{\gamma}_i \bar{\beta}_{ij}, \end{aligned}$$

We then obtain:

$$\begin{aligned} & \left[1 + \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k)\right] e_i(t_{k+1}) \\ = \ & \hat{\gamma}_i \left[1 + \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k)\right] x_i(t_k) + \sum\nolimits_j \hat{\alpha}_{ij} x_j^{n_{ij}}(t_k) + C_i \end{aligned}$$

This leads to

$$\begin{aligned}
& e_i(t_{k+1}) \\
= \; & \hat{\gamma}_i x_i(t_k) + \sum\nolimits_j \hat{\alpha}_{ij} x_j^{n_{ij}}(t_k) + \sum\nolimits_j \hat{\beta}_{ij} x_j^{n_{ij}}(t_k) x_i(t_k) \\
& - \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k) e_i(t_{k+1}) + C_i, \\
= \; & \hat{\gamma}_i x_i(t_k) + \sum\nolimits_j \hat{\alpha}_{ij} x_j^{n_{ij}}(t_k) + \sum\nolimits_j \hat{\beta}_{ij} x_j^{n_{ij}}(t_k) x_i(t_k) \\
& - \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k)(x_i(t_{k+1}) - x_i(t_k)) + C_i \\
= \; & \hat{\gamma}_i x_i(t_k) + \sum\nolimits_j \hat{\alpha}_{ij} x_j^{n_{ij}}(t_k) + \sum\nolimits_j \hat{\beta}_{ij} x_j^{n_{ij}}(t_k) x_i(t_k) \\
& - \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k) x_i(t_{k+1}) + \sum\nolimits_j \bar{\beta}_{ij} x_j^{n_{ij}}(t_k) x_i(t_k) + C_i,
\end{aligned}$$

from which $\hat{\gamma}_i, \hat{\alpha}_{ij}, \hat{\beta}_{ij}, \bar{\beta}_{ij}, C_i$ can be estimated. We can then calculate $\gamma_i, \bar{\beta}_{ij}$ and $\bar{\alpha}_{ij}$.

2) Repression Hill Functions: $n_{ij} = 0$: We let

$$\begin{aligned}
& \tilde{\gamma}_i = -\gamma_i, \quad \tilde{\alpha}_{ij} = \alpha_{ij}, \quad \tilde{\beta}_{ij} = \beta_{ij}/K_{ij}^{m_{ij}}, \\
& \check{\alpha}_{ij} = C_i \tilde{\beta}_{ij}, \quad \check{\beta}_{ij} = \tilde{\gamma}_i \tilde{\beta}_{ij}, \quad \tilde{C}_i = \sum\nolimits_j \tilde{\alpha}_{ij} + C_i.
\end{aligned}$$

We then obtain:

$$\begin{aligned}
& \left[1 + \sum\nolimits_j \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k)\right] e_i(t_{k+1}) \\
= \; & \tilde{\gamma}_i \left[1 + \sum\nolimits_j \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k)\right] x_i(t_k) + \sum\nolimits_j C_i \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k) + \tilde{C}_i
\end{aligned}$$

This leads to:

$$\begin{aligned}
& e_i(t_{k+1}) \\
= \; & \tilde{\gamma}_i x_i(t_k) + \sum\nolimits_j \check{\alpha}_{ij} x_j^{m_{ij}}(t_k) + \sum\nolimits_j \check{\beta}_{ij} x_j^{m_{ij}}(t_k) x_i(t_k) \\
& - \sum\nolimits_j \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k)(x_i(t_{k+1}) - x_i(t_k)) + \tilde{C}_i \\
= \; & \tilde{\gamma}_i x_i(t_k) + \sum\nolimits_j \check{\alpha}_{ij} x_j^{m_{ij}}(t_k) + \sum\nolimits_j \check{\beta}_{ij} x_j^{m_{ij}}(t_k) x_i(t_k) \\
& - \sum\nolimits_j \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k) x_i(t_{k+1}) + \sum\nolimits_j \tilde{\beta}_{ij} x_j^{m_{ij}}(t_k) x_i(t_k) + \tilde{C}_i
\end{aligned}$$

from which $\tilde{\gamma}_i, \check{\alpha}_{ij}, \check{\beta}_{ij}, \tilde{\beta}_{ij}, \tilde{C}_i$ can be estimated. Based on these, we can then calculate $\gamma_i, \bar{\beta}_{ij}$ and $\bar{\alpha}_{ij}$.

We thus see that the problem of identification of rational polynomial functions with unknown parameters in their denominator can be reduced to a problem of polynomial identification. This can be done following the approach presented in Section III-B by extending the basis functions.

To consider various Hill coefficients m_j, e.g., all possible Hill coefficient values from the set $\{1, 2, 3, 4\}$, we just need to consider all the functions $G_j(x)$ with these individual values of m_j and to add the corresponding monomial terms extracted from the above derivations as elements of the sensing matrix Φ.

D. Problem Transformation

Once the matrix Φ is constructed so as to contain all the candidate basis functions that we want to consider in the reconstruction, the remaining task is to propose an efficient method that allows to solve for $\mathbf{w}_i$ in eq. (10):

$$\mathbf{y}_i = \Phi \mathbf{w}_i + \xi_i, \ (i = 1, \ldots, n).$$

Typically the weighting vector $\mathbf{w}_i$ solution of (10) is *k-sparse.* Mathematically, we say that a signal $\mathbf{w}$ is *k-sparse* when it has at most *k* non-zero entries, i.e., $\|\mathbf{w}\|_0 \leq k$. We let

$$\Omega_k = \{\mathbf{w} : \|\mathbf{w}\|_0 \leq k\}$$

denote the set of all k-sparse signals. On one hand, biochemical reaction networks are typically sparse [13]. On the other hand, the acquisition of sufficient biological time series data over long time spans is quite difficult due to the typical cost of wet-lab experiments and current technological limitations in terms of the type and quality of the measurements. Furthermore, since the nonlinear form of the equation is typically unknown, there can potentially be a very large number of candidate functions. As a consequence, we typically have $N \gg M$ for $\Phi_{M\times N}$ and $\mathbf{w}_i$ sparse.

The linear regression problem (10) can thus be defined as a compressive sensing, or sparse signal recovery problem [6], [10], with observation vector $\mathbf{y}_i$, known regressor matrix Φ, unknown coefficients $\mathbf{w}_i$, and additive noise Ξ. In sparse problems, the prior belief is that only a small fraction of the elements appearing in $\mathbf{w}_i$ are non-negligible. The general aim is to identify the smallest subset of columns of Φ, whose linear span contains the observations $\mathbf{y}_i$. Identifying this smallest subset for all $i = 1, \ldots, n$ corresponds to looking for the sparsest $\mathbf{w}$ that satisfies the following equation:

$$\mathbf{y} = \Phi \mathbf{w} + \Xi. \tag{14}$$

The solution $\mathbf{w}^*$ to eq. (14) is, by definition, going to be sparse. Now, if all the elements in the i^{th} row of $\mathbf{w}^*$ are 0, then we can delete the i^{th} row of $\mathbf{w}^*$ and the i^{th} column of Φ. After repeating this for all the rows of $\mathbf{w}^*$, we can equivalently rewrite (14) as

$$\mathbf{y} = \Phi^* \bar{\mathbf{w}}^* + \Xi^*. \tag{15}$$

The solution of eq. (15) is an estimate of the solution of the original problem defined in eq. (7) which itself is related to the discretisation of eq. (1).

IV. COMPRESSIVE SENSING

A. Algorithm for Compressive Sensing

To be consistent with eq. (10), in this and next section, $i = 1, \ldots, n$. Since $\mathbf{w}_i$ is sparse, we use regularisation methods to recover $\mathbf{w}_i$. Regularisation methods impose upper bound constraints on the l_0 norm of $\mathbf{w}_i$, $\|\mathbf{w}_i\|_0$, i.e., on the number of nonzero terms in $\mathbf{w}_i$. A sparse approximation to $\mathbf{w}_i$ is readily obtained by solving the following optimisation problem:

$$\hat{\mathbf{w}}_\mathbf{i} = \arg\min_{\mathbf{w}_i} \{\|\mathbf{y}_i - \Phi \mathbf{w}_i\|_2^2 + \rho_i \|\mathbf{w}_i\|_0\}. \tag{16}$$

Unfortunately, this optimisation problem is both numerically unstable and NP-complete; therefore some simplifications are typically used to recast this problem into another one for which efficient algorithmic solutions exist. The most common simplification is to use the l_1-norm instead of the l_0-norm [6], so that the optimisation problem becomes

$$\hat{\mathbf{w}}_{\mathbf{i}} = \arg\min_{\mathbf{w}_i}\{\|\mathbf{y}_i - \Phi\mathbf{w}_i\|_2^2 + \rho_i \|\mathbf{w}_i\|_1\}. \tag{17}$$

The approach presented in eq. (17) is known as *Lasso* [23]. A number of methods have been proposed to solve Compressive Sensing (CS) reconstruction problems, including l_1-minimisations [6], [7] and greedy algorithms [8], [11], [27]. Both methods can actually be shown to have performance guarantees for the exact recovery of $\mathbf{w}_i$ if Φ satisfies the so called *restricted isometry property* [6]. We will discuss this in more detail in Section VII.

Since Φ as defined in (9) could be rank deficient, (17) cannot, in general, be readily solved by these two algorithms and thus a different approach needs to be considered. Thanks to recent results in signal processing and machine learning, we propose hereafter a method to solve the compressive sensing problem defined in (17) using a Bayesian formulation [14], [24]–[26].

B. Bayesian Compressive Sensing

1) Linear Regression and Maximum Likelihood: Bayesian modelling treats all unknowns as stochastic variables with certain probability distributions [3]. Using a Bayesian framework, we assume that the stochastic variable ξ_i are independent and characterised by a Gaussian distribution with zero-mean and variance σ_i^2. We further define the precision or inverse-variance as $\beta_i = 1/\sigma_i^2$.

The multivariate Gaussian likelihood of $\mathbf{y}_i$ is then given by:

$$\begin{aligned} p(\mathbf{y}_i|\mathbf{w}_i,\beta_i) &= \mathcal{N}(\mathbf{y}_i|\Phi\mathbf{w}_i,\beta_i^{-1}) \\ &= (2\pi\sigma_i^2)^{-\frac{M}{2}} \exp\left(-\frac{1}{2\sigma_i^2}\|\mathbf{y}_i - \Phi\mathbf{w}_i\|^2\right) \end{aligned} \tag{18}$$

Obtaining maximum likelihood estimates for $\mathbf{w}_i$ under these conditions is equivalent to searching for a minimal l_2-norm solution to eq. (7).

2) Sparseness Prior: The sparseness of $\mathbf{w}_i$ can be imposed by using a Laplace prior of the form

$$p(\mathbf{w}_i|\lambda) = \frac{\lambda_i}{2}\exp(-\lambda_i \|\mathbf{w}_i\|_1). \tag{19}$$

This then becomes equivalent to the l_1-regularisation formulation in eq. (17). Using the likelihood function in eq. (18) and the prior in eq. (19), the solution of eq. (17) can be shown to correspond to a *maximum a posteriori* (MAP) estimate for $\mathbf{w}_i$.

However, obtaining an estimate of the full posteriors on $\mathbf{w}_i$ and β_i cannot be accomplished using the Laplace prior directly since Laplace priors are not conjugate to the conditional Gaussian distribution in eq. (18). In the following section, we show that the use of hierarchical priors alleviates this. The CS problem can thus be converted into a linear-regression problem with a prior which is sparse. Given the sensing matrix Φ, we show in the next section how the sparse weights $\mathbf{w}_i$ and the inverse of noise variance β_i can be estimated using in a sparse Bayesian learning approach.

V. RECONSTRUCTION VIA BAYESIAN COMPRESSIVE SENSING

A. *Specification of Hierarchical Priors*

Instead of imposing a Laplace prior on $\mathbf{w}_i$, as defined in the last section, a sparse Bayesian learning approach is adopted [25]. Namely, we use hierarchical priors over the distribution of $\mathbf{w}_i$. This allows for a convenient conjugate-exponential analysis with similar properties as for the Laplace priors. The main advantage of such an approach is that it allows us to impose a Gaussian prior with zero-mean on each element of $\mathbf{w}_i$ (see (11)), i.e.,

$$p(\mathbf{w}_i|\boldsymbol{\alpha}_i) = \prod_{j=1}^{N} \mathcal{N}(\omega_{ij}|0, \alpha_{ij}^{-1}). \tag{20}$$

In (20), $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iN}) \in \mathbb{R}^{1\times N}$ represent a vector of N independent hyperparameters, with α_{ij} controlling the precision (or the inverse of the variance) of the prior imposed on $\mathbf{w}_i$. It is this form of prior that is eventually responsible for the sparsity properties of the model (see [25] for more details). It is common to place a Gamma prior on $\boldsymbol{\alpha}_i$ as follows:

$$p(\boldsymbol{\alpha}_i|a, b) = \prod_{j=1}^{N} \Gamma(\alpha_{ij}|a, b). \tag{21}$$

where the Gamma distribution is defined as:

$$\Gamma(\xi|a, b) = \frac{(b)^a}{\Gamma(a)} \xi^{a-1} \exp[-b\xi]$$

where $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$ is called the ‘Gamma function’, $\xi > 0$ denotes a hyperparameter, and $a > 0$ is the shape parameter, $b > 0$ is a scaling parameter. The Γ distribution is generally chosen as the prior for the precision of a Gaussian distribution because (a) it corresponds to its conjugate prior, thereby greatly simplifying the analysis and (b) it also includes the uniform distribution as a limiting case. The overall prior on $\mathbf{w}_i$ is then evaluated as

$$p(\mathbf{w}_i|a, b) = \prod_{j=1}^{N} \int_0^\infty \mathcal{N}(\omega_{ij}|0, \alpha_{ij}^{-1}) \Gamma(\alpha_{ij}|a, b) d\alpha_{ij}. \tag{22}$$

The density function $\Gamma(\alpha_{ij}|a, b)$ is the conjugate prior for α_{ij} when ω_{ij} plays the role of observed data and $\mathcal{N}(\omega_{ij}|0, \alpha_{ij}^{-1})$ is the associated likelihood function. Based on this, the integral $\int_0^\infty \mathcal{N}(\omega_{ij}|0, \alpha_{ij}^{-1}) \Gamma(\alpha_{ij}|a, b) d\alpha_{ij}$ can be evaluated analytically. It can be shown that this integral corresponds to the Student’s *t*-distribution which is strongly peaked around $\omega_{ij} = 0$. Consequently, the prior in (22) is a sparseness prior for $\mathbf{w}_i$. Similarly, a Gamma prior is introduced on β_i

$$p(\beta_i|c, d) = \Gamma(\beta_i|c, d). \tag{23}$$

B. *Bayesian Inference via Relevance Vector Machine*

Given $\mathbf{y}_i$ and $\boldsymbol{\Phi}$ and assuming that the hyperparameters $\boldsymbol{\alpha}_i$ and β_i are known, the *posterior* distribution for $\mathbf{w}_i$ conditioned on the data is given by combining the likelihood and prior with Bayes’ rule:

$$p(\mathbf{w}_i|\mathbf{y}_i, \boldsymbol{\alpha}_i, \beta_i) = \frac{p(\mathbf{y}_i|\mathbf{w}_i, \beta_i) p(\mathbf{w}_i|\boldsymbol{\alpha}_i)}{p(\mathbf{y}_i|\boldsymbol{\alpha}_i, \beta_i)}. \tag{24}$$

This posterior distribution of $\mathbf{w}_i$ is Gaussian and the associated mean and covariance matrices are given as

$$\begin{aligned}\mathbf{m}_i &= \beta_i \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y}_i && (25)\\ \boldsymbol{\Sigma}_i &= (\mathbf{A}_i + \beta_i \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} && (26)\end{aligned}$$

where $\mathbf{A}_i = \mathrm{diag}(\alpha_{ij})$. In the context of Relevance Vector Machines (RVM), the associated "learning" problem becomes the search for the hyperparameters $\boldsymbol{\alpha}_i$ and β_i. RVM is a Bayesian sparse kernel technique for regression and classification [25]. It shares a lot of characteristics with support vector machines [3]. In RVM, these hyperparameters are estimated from the data by maximising

$$p(\mathbf{y}_i|\boldsymbol{\alpha}_i, \beta_i) = \int p(\mathbf{y}_i|w_i, \beta_i) p(\mathbf{w}_i|\boldsymbol{\alpha}_i) d\mathbf{w}_i.$$

This quantity is known as the marginal likelihood. The maximisation is known as a type-2 maximum likelihood [2] or evidence approximation maximisation [16].

To avoid the convolution of two Gaussians, one can use log marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}_i, \beta_i)$. This leads to

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}_i, \beta_i) &= \log p(\mathbf{y}_i|\boldsymbol{\alpha}_i, \beta_i)\\ &= \log \int p(\mathbf{y}_i|w_i, \beta_i) p(\mathbf{w}_i|\boldsymbol{\alpha}_i) d\mathbf{w}_i\\ &= \log \mathcal{N}(\mathbf{y}_i|0, \mathbf{C}_i)\\ &= -\frac{1}{2}\left[M \log 2\pi + \log|\mathbf{C}_i| + \mathbf{y}_i^T \mathbf{C}_i^{-1} \mathbf{y}_i\right]\end{aligned} \tag{27}$$

where the $M \times M$ matrix $\mathbf{C}_i = \sigma_i^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}_i^{-1} \boldsymbol{\Phi}^T$. A type-2 maximum likelihood approximation employs the point estimates for $\boldsymbol{\alpha}_i$ and β_j to maximise $\mathcal{L}(\boldsymbol{\alpha}_i, \beta_j)$. By setting the required derivatives of the marginal likelihood to zero, we can obtain the following re-estimate equations [16], [25]

$$\alpha_{ij}^{new} = \frac{\gamma_{ij}}{m_{ij}^2}, (j = 1, 2, \ldots, N), \tag{28}$$

where $\mathbf{m}_i$ is the ith posterior mean weight appearing in (25) and $\gamma_{ij} \triangleq 1 - \alpha_{ij} \boldsymbol{\Sigma}_i(jj)$, with $\boldsymbol{\Sigma}_i(jj)$ representing the jth diagonal element of the posterior weight covariance defined in (26). Estimation of γ_{ij} and m_{ij} can be efficiently done via the expectation-maximisation algorithm, an iterative procedure for maximum likelihood parameter estimation from data sets with missing or hidden variables [9].

For a noise variance $\sigma_i^2 = 1/\beta_i$, the re-estimate equation is

$$(\beta_i^{new})^{-1} = \frac{\|\mathbf{y}_i - \boldsymbol{\Phi} \mathbf{m}_i\|_2^2}{M - \sum_j \gamma_{ij}}. \tag{29}$$

The quantity γ_{ij} in (29) measures how well the corresponding parameter ω_{ij} appearing in (10) is determined by the data. Note that α_{ij}^{new}, $j = 1, \ldots, N$ and β_i^{new} are functions of $\mathbf{m}_i$ and $\boldsymbol{\Sigma}_i$ in equations (28)-(29). Furthermore, $\mathbf{m}_i$ and $\boldsymbol{\Sigma}_i$ are a function of $\boldsymbol{\alpha}_i$ and β_i in equations (25)-(26). This suggests an iterative algorithm, which iterates between equations (25)-(26) and equations (28)-(29) until a convergence criterion has been satisfied. During re-estimation, many of the α_{ij} tend to infinity when the corresponding ω_{ij} are very small or zero. Usually, only a few α_{ij} are small. From (24), this implies that $p(\omega_{ij}|\mathbf{y}_i, \boldsymbol{\alpha}_i, \beta_i)$ becomes highly (in principle, infinitely) peaked at zero. This implies that we can be *a posteriori* 'certain' that ω_{ij} are zero. The corresponding basis functions in Φ can thus be 'pruned', and sparsity is realised [25]. In [24] and [26], a fast RVM algorithm was developed by analysing the properties of the log marginal likelihood function in eq. (27). This enables a principled and efficient sequential addition and deletion of candidate basis function (columns of $\boldsymbol{\Phi}$) to monotonically maximise the log marginal likelihood. Details about this fast algorithm can be found in [24] and [26].

VI. IDENTIFICATION OF THE REPRESSILATOR

We consider here a classical dynamical system in systems/synthetic biology which we will use to illustrate the biochemical network reconstruction problem at hand. The repressilator is a synthetic oscillator network that was originally conceived and constructed by Elowitz and Leibler [12]. The network consists of three genes repressing each other in a ring structure.

A mathematical description the repressilator that includes both transcription and translation dynamics can be described as

$$\begin{aligned}
\frac{dx_1}{dt} &= -\gamma_1 x_1 + \frac{\alpha_1}{1+x_6^{n_1}} + \theta_1, \\
\frac{dx_2}{dt} &= -\gamma_2 x_2 + \frac{\alpha_2}{1+x_4^{n_2}} + \theta_2, \\
\frac{dx_3}{dt} &= -\gamma_3 x_3 + \frac{\alpha_3}{1+x_5^{n_3}} + \theta_3, \\
\frac{dx_4}{dt} &= -\gamma_4 x_4 + \beta_1 x_1, \\
\frac{dx_5}{dt} &= -\gamma_5 x_5 + \beta_2 x_2, \\
\frac{dx_6}{dt} &= -\gamma_6 x_6 + \beta_3 x_3,
\end{aligned} \tag{30}$$

Here, x_1, x_2, x_3 denote the concentrations of the mRNA transcripts of genes 1, 2, and 3, respectively whereas x_4, x_5, x_6 denote the protein concentrations of the respective genes. $\alpha_1, \alpha_2, \alpha_3$ denote the maximum promoter strength for their corresponding gene, $\gamma_1, \gamma_2, \gamma_3$ denote the mRNA decay rate, $\gamma_4, \gamma_5, \gamma_6$ denote the protein decay rate, $\beta_1, \beta_2, \beta_3$ denote the protein production rate, $\theta_1, \theta_2, \theta_3$ denote the basal transcription rate. The set of ODEs in (30) corresponds to a topology where gene 1 is repressed by gene 2, gene 2 is repressed by gene 3, and gene 3 is repressed by gene 1. Using the standard forward Euler method to numerically solve ODEs, we obtain trajectory of the six states $x_1, \ldots, x_6$ in Fig. 1. These trajectories are then sampled to generate a time-series of gene expression data.

Take gene 1 for example. The hill coefficient n_1 will typically have a value within a range from 1 to 4 due to biochemical constraints. The core question here is: how can we determine the topology and kinetic parameters of the set of ODEs in (30)? Note that we do not assume *a priori* knowledge of the form of the nonlinear functions, e.g., whether the degradation obeys first-order or enzymatic catalysed dynamics or whether the proteins are repressors or activators.

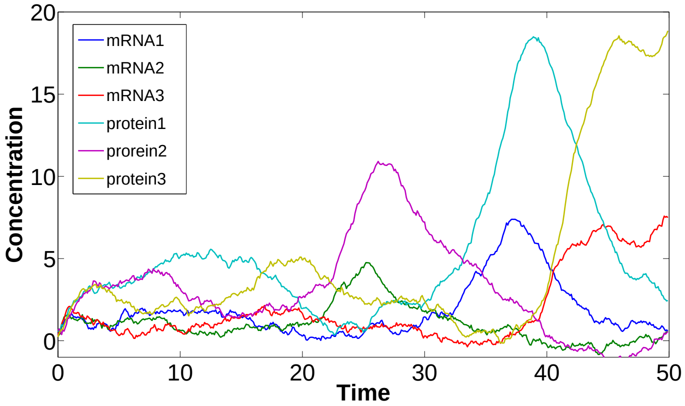

Fig. 1. Time series of $x_1, \dots, x_6$ for the repressilator model (30).

$$\Phi = \begin{bmatrix} x_1(t_1) & \dots & x_6(t_1) & \text{hill}_1(t_1) & \dots & \text{hill}_4(t_1) & 1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ x_1(t_{K-1}) & \dots & x_6(t_{K-1}) & \text{hill}_1(t_{K-1}) & \dots & \text{hill}_4(t_{K-1}) & 1 \end{bmatrix}_{(K-1)\times(6+48+1)} \tag{31}$$

Next we show how the network construction problem of the repressilator model in (30) can be formulated under the form presented in (10).

Following the procedure in (2), (3) and (6), we construct a matrix of candidate functions Φ by selecting the most commonly used candidate basis functions used to model biochemical reaction networks (see Sections III-B and III-C). As a proof of concept, we only consider Hill functions as potential nonlinear candidates. The set of Hill functions with Hill coefficient i, both in activating and repressing from, for each of the 6 state variables reads:

$$\begin{aligned} \text{hill}_i(t_k) \quad \triangleq \quad & \left[\frac{1}{1+x_1^i(t_k)}, \dots, \frac{1}{1+x_6^i(t_k)}, \right. \\ & \left. \frac{x_1^i(t_k)}{1+x_1^i(t_k)}, \dots, \frac{x_6^i(t_k)}{1+x_6^i(t_k)} \right]_{1\times 12}, \end{aligned}$$

where i represents the Hill coefficient. In what follows we consider that the Hill coefficient can take any of the following values: 1, 2, 3 or 4. To also take into account basal transcription expression rates, we add to the last column of Φ a unit vector. Since there are 6 state variables, we can construct the basis matrix Φ appearing in (31) with 6 (basis functions for linear terms) $+(4*12) = 48$ (basis functions for hill functions) $+1$ (basis function for basal expression) $= 55$ columns.

Considering (10) with the basis function matrix Φ given in (31), the corresponding target weight matrix $\mathbf{w}$ should be:

$$\begin{bmatrix} -\gamma_1 & 0 & 0 & \beta_1 & 0 & 0 \\ 0 & -\gamma_2 & 0 & 0 & \beta_2 & 0 \\ 0 & 0 & -\gamma_3 & 0 & 0 & \beta_3 \\ 0 & 0 & 0 & -\gamma_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma_6 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ \alpha_1 & \alpha_2 & \alpha_3 & 0_{48\times1} & 0_{48\times1} & 0_{48\times1} \\ 0_{30\times1} & 0_{32\times1} & 0_{31\times1} & & & \\ \theta_1 & \theta_2 & \theta_3 & 0 & 0 & 0 \end{bmatrix}.$$

As an illustration, we choose the parameter values for $\mathbf{w}$ as:

$$\begin{bmatrix} -0.3 & 0 & 0 & 1.4 & 0 & 0 \\ 0 & -0.4 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 1.6 \\ 0 & 0 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.6 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ 4 & 3 & 5 & 0_{48\times1} & 0_{48\times1} & 0_{48\times1} \\ 0_{30\times1} & 0_{32\times1} & 0_{31\times1} & & & \\ 0.02 & 0.02 & 0.01 & 0 & 0 & 0 \end{bmatrix}.$$

The algorithm is implemented in MATLAB R2009a. The calculation is run on a standard laptop computer (Intel Core2 Duo P8600 2.4GHz with 4GB RAM). We run the algorithm for the following two experiments independently with 100 rounds respectively. The sparsest and closest solutions were selected. The computation time for each round is less than **0.1** second.

1) Let $t_T = 50$, sampling interval $t_{k+1} - t_k = 0.1$, and $T = 500$. The variance of noise is fixed at 10^{-3}. The estimated $\mathbf{w}$ is

$$\begin{bmatrix} 0.300 & 0 & 0 & 1.401 & 0 & 0 \\ 0 & -0.398 & 0 & 0 & 1.502 & 0 \\ 0 & 0 & -0.499 & 0 & 0 & 1.600 \\ 0 & 0 & 0 & -0.200 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.400 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.600 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ 3.999 & 3.002 & 5.001 & 0_{48\times1} & 0_{48\times1} & 0_{48\times1} \\ 0_{30\times1} & 0_{32\times1} & 0_{31\times1} & & & \\ 0.019 & 0.019 & 0.0093 & 0 & 0 & 0 \end{bmatrix}.$$

2) Let $t_T = 5$, sampling interval $t_{k+1} - t_k = 0.1$, and $T = 50$. The variance of noise is fixed at 10^{-3}. The estimated $\mathbf{w}$ is

$$\begin{bmatrix} -0.296 & 0 & 0 & 1.389 & 0 & 0 \\ 0 & -0.395 & 0 & 0 & 1.511 & 0 \\ 0 & 0 & -0.500 & 0 & 0 & 1.607 \\ 0 & 0 & 0 & -0.197 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.406 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.601 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ 4.066 & 3.033 & 5.028 & 0_{48\times1} & 0_{48\times1} & 0_{48\times1} \\ 0_{30\times1} & 0_{32\times1} & 0_{31\times1} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the basal rate could not be estimated in the lesser sample case.

VII. CONCLUSION AND DISCUSSION

In this paper, a new network reconstruction method for biochemical reaction networks is proposed. This method takes advantage of compressive sensing and sparse Bayesian learning. The proposed method only requires time series data and does not assume prior knowledge about the model structure (topology and nonlinear functional forms) and parameters. The problem is posed in such a way that candidate nonlinear functions specific to the type of models used (here biochemical reaction networks) are sought after. Since biochemical reaction networks are typically sparse, the key idea is to adopt a formulation which allows to transform the nonlinear identification problem into a compressive sensing problem and to solve it efficiently using a sparse Bayesian learning approach. We have illustrated how this approach can be used to efficiently reconstruct the nonlinear ODEs of a repressilator based on time series data.

However, beyond the results presented here, some issues remain to be further discussed and are part of ongoing work in our group.

A. Performance Guarantee

A sufficient condition for exact reconstruction with either l_1-minimisation or greedy algorithms is the so called *restricted isometry property* (RIP): A matrix $\Phi \in \mathbb{R}^{M \times N}$ is said to satisfy the RIP with coefficients (K, δ) for $K \leq M$, $0 \leq \delta \leq 1$, if for all index sets $I \subset \{1, \ldots, N\}$ such that $|I| \leq K$ and for all $q \in \mathbb{R}^{|I|}$, one has

$$(1 - \delta) \|q\|_2^2 \leq \|\Phi_i q\|_2^2 \leq (1 + \delta) \|q\|_2^2$$

where Φ_i denotes the matrix formed by the columns of Φ with indices in I. It was shown in [5], [6], [8] that both l_1-minimisations and greedy algorithms lead to exact reconstruction of K-sparse signals if the matrix Φ satisfies the RIP with a constant parameter $0 \leq \delta \leq 1$.

However, the Bayesian method proposed in this paper has no known performance guarantee in terms of exact recovery equivalent to the RIP condition. Establishment of such performance guarantees in the Bayesian compressive sensing framework is part of ongoing work in our group.

B. Hidden Nodes

We have so far assumed that the system is fully observable. However, in reality, measurement observations will typically be partial [28] (in particular, the number of hidden/unobservable nodes and their position in the network is usually unknown). How to generalise our framework to the case of hidden nodes and partial observations is another topic of further study in our group.

VIII. ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of Microsoft Research through the PhD Scholarship Programs of Mr Wei Pan and Dr Ye Yuan. Dr Ye Yuan acknowledges the support from EPSRC (project EP/I03210X/1). Dr Guy-Bart Stan gratefully acknowledges the support of the EPSRC Centre for Synthetic Biology and Innovation at Imperial College, London. The authors would like to thank Dr Wei Dai (at Imperial College London) for the constructive discussions that lead to this paper.

REFERENCES

[1] E. August, and A. Papachristodoulou, Efficient, sparse biological network determination, *BMC Systems Biology*, vol. 3(1), 2009, pp. 25.

[2] J. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985.

[3] C. Bishop. *Pattern recognition and machine learning*, Springer New York; 2006.

[4] N. Buchler, U. Gerland, and T. Hwa. On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences*, vol. 100(9), 2003, pp. 5136.

[5] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8), 2006, pp. 1207-1223.

[6] E. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, vol. 51(12), 2005, pp. 4203-4215.

[7] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, vol. 20(1), 1999, pp. 33-61.

[8] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, vol. 55(5), 2009, pp. 2230-2249.

[9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977, pp. 1-38.

[10] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, vol. 52(4), 2006, pp. 1289-1306.

[11] D. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102(27), 2005, pp. 9446.

[12] M. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, vol. 403(6767), 2000, pp. 335-8.

[13] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. The large-scale organization of metabolic networks. *Nature*, vol. 407(6804), 2000, pp. 651-654.

[14] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, vol. 56(6), 2008, pp. 2346-2356.

[15] L. Ljung, H. Hjalmarsson, and H. Ohlsson. Four encounters with system identification. *European Journal of Control*, vol. 17(5), 2011. pp. 449.

[16] D. MacKay. Bayesian interpolation. *Neural computation*, vol. 4(3), 1992, pp. 415-447.

[17] F. Montefusco, C. Cosentino, and F. Amato, CORENet: exploiting prior knowledge and preferential attachment to infer biological interaction networks, *IET Systems Biology*, vol. 4, 2010, pp. 296-310.

[18] R. Porreca, E. Cinquemani, J. Lygeros, and G. FerrariTrecate, Identification of genetic network dynamics with unate structure, *Bioinformatics*, vol. 26(9), 2010, pp. 123C129.

[19] B. Palsson. *Systems biology, pp properties of reconstructed networks*. Cambridge University Press, 2006.

[20] B. Sanandaji, T. Vincent and M. Wakin, Exact Topology Identification of Large-Scale Interconnected Dynamical Systems from Compressive Observations, *Proceedings of the American Control Conference*, 2011.

[21] Y. Setty, A. Mayo, M. Surette, and U. Alon. Detailed map of a cis-regulatory input function. *Proceedings of the National Academy of Sciences*, vol. 100(13), 2003, pp. 7702.

[22] E. Sontag, A. Kiyatkin, and B. Kholodenko, Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data, *Bioinformatics*, vol. 20(12), 2004, pp. 1877-1886.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996, pp. 267-288.

[24] A. Tipping and A. Faul. Analysis of sparse bayesian learning. *Advances in Neural Information Processing Systems*, vol. 14, pp. 383-389,2002.

[25] M. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, vol. 1, 2001, pp. 211-244.

[26] M. Tipping, A. Faul, et al. Fast marginal likelihood maximisation for sparse bayesian models. In *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1, 2003.

[27] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, vol. 53(12), 2007, pp. 4655-4666.

[28] Y. Yuan, G. Stan, S. Warnick, and J. Goncalves. Robust dynamical network structure reconstruction. *Automatica*, vol. 47, 2011, pp. 1230-1235.

[29] M. Zavlanos, A. Julius, S. Boyd, and G. Pappas, Inferring stable genetic networks from steady-state data, *Automatica*, vol. 47, 2011, pp. 1113-1122