Reconstruction of Arbitrary Biochemical Reaction Networks: A Compressive Sensing Approach

Wei Pan, Ye Yuan, Jorge Gonçalves and Guy-Bart Stan[†]

Abstract— **Reconstruction of biochemical reaction networks (BRN) and genetic regulatory networks (GRN) in particular 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 ODE model descriptions containing 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 parameters. To solve the network reconstruction problem, we cast it as a compressive sensing (CS) problem and use sparse Bayesian learning (SBL) algorithms as a computationally efficient and robust 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 parameters of BRN. 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 data. The associated computational burden of such an approach is typically horrendous even for network of modest dimensions. Within the systems biology and control community, identification of BRN and GRN in particular, are quite active research areas [1]–[4].

Many linear and nonlinear functions can be used to describe the dynamics of BRN in terms of biochemical kinetic laws, e.g., 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, $i = 1,\ldots,n$, for example, $f(x_1,\ldots,x_n) = \sum_i \beta_i(x_i/K_i)^{n_i}/(1+\sum_i \beta_i(x_i/K_i)^{m_i})$, 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 types of functions have been shown to appropriately describe experimentally determined input functions [5].

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. The major two of these were: (a) the use of sparsity and (b) the use of machine learning approaches [6]. 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 CS and SBL together to offer an efficient method for biological network reconstruction.

The paper is organised as follows. In Section II, we introduce the type of BRN model we consider in this paper. 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 CS problem. In Section V, we show how SBL algorithms can be used to solve the CS 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 CS framework. Finally, in Section VII, we conclude and discuss several future problems that we plan to address.

II. MODEL FORMULATION

We consider dynamical systems described by nonlinear ODEs with additive noise:

$$e = g(x) + \xi, \tag{1}$$

where $e = [e_1, e_2, \ldots, e_n]^T \in \mathbb{R}^n$ are the system responses; $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ denotes the state vector; and $\xi \in \mathbb{R}^n$ denotes a vector of additive noise. In continuous-time systems, e denotes the derivative of the state variables, i.e. $e = \dot{x} = dx/dt$ and eq. (1) thus becomes: $\frac{dx(t)}{dt} = g(x(t)) + \xi(t)$. In discrete-time systems, e represents the value of the state variables at the next discrete-time step,

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 and J. Gonçalves are with Control Group, Department of Engineering, University of Cambridge, United Kingdom.

[†]For correspondence, g.stan@imperial.ac.uk.

i.e., $e = x(t_{k+1})$. Then eq. (1) becomes: $x(t_{k+1}) = g(x(t_k)) + \xi(t_k)$. Since biochemical reaction are typically governed by mass action kinetics, Michaelis-Menten, or Hill kinetics, $g(x)$ belongs to a certain set of functions of known form, e.g., mass action kinetic terms under the form of product of monomials, monotonically increasing or monotonically decreasing Hill functions, simple linear terms, constant terms, etc. The nonlinear function $g(x)$ can thus be decomposed into a linear sum of basis functions $f_i(x)$, e.g. $g(x) = \sum_{i=1}^{L} v_i f_i(x)$. Measuring/estimating the time derivative from noisy data in continuous-time systems can either be achieved using a measurement equipment with a sufficiently high sampling rate, or be estimated using state-of-the-art mathematical approaches [7].

Suppose the time series data are sampled from a real experimental system at discrete time points t_k. For BRN with stoichiometric matrix $\mathbf{S}$, eq. (1) can be written under the form [8]:

$$e(t_{k+1}) = \mathbf{S}f(x(t_k)) + \xi(t_k), \tag{2}$$

where $\mathbf{S} = \begin{bmatrix} v_{11} & \dots & v_{L1} \\ \vdots & \vdots & \vdots \\ v_{1n} & \dots & v_{Ln} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix} \in \mathbb{R}^{n \times L}$ denotes the stoichiometry matrix; $f(x) = [f_1(x), f_2(x), \dots, f_L(x)]^T \in \mathbb{R}^L$ is the vector field; and $\xi = [\xi_1, \xi_2, \dots, \xi_n]^T \in \mathbb{R}^n$ represents energy-bounded process noises which are assumed to be independent and to be distributed according to normal probability distributions: $E[\xi(t_p)] = 0$, $E[\xi(t_p)\xi^T(t_q)] = Q_{pq}\delta_{pq}$, where $\delta_{pq} = \begin{cases} 1, & p = q, \\ 0, & p \neq q. \end{cases}$ From experimental data given as time series of both e and x, our objective is to identify the stoichiometry coefficient in $\mathbf{S}$. In the following section, we shall propose an algorithm that uses time-series data to reconstruct eq. (2).

III. RECONSTRUCTION PROBLEM FORMULATION

A. Problem Formulation

Taking the transpose of both sides of eq. (2), we obtain:

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

where $\mathbf{v} = \mathbf{S}^T = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{L \times n}$. Assuming that M successive data points are sampled and defining

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

eq. (3) can be written as n independent equations:

$$\mathbf{y}_i = \mathbf{\Theta}\mathbf{v}_i + \mathbf{\Xi}_i, \ (i = 1, \dots, n). \tag{4}$$

We want to find $\mathbf{v}_i$ given the 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 $\mathbf{\Theta}$ is known. However, in most cases, these nonlinearities are unknown or difficult to assume *a priori*. As a consequence, $\mathbf{\Theta}$ itself is unknown and, therefore, $\mathbf{v}_i$ cannot be solved from eq. (4).

Fortunately, some *a priori* knowledge of the field in which the models are developed can help. Indeed, 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 BRN 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 $\mathbf{\Phi}$ similar to $\mathbf{\Theta}$ the set of ***all*** candidate basis functions that we want to consider for reconstruction:

$$\mathbf{\Phi} \triangleq \begin{bmatrix} F_1(x(t_0)) & \dots & F_N(x(t_0)) \\ F_1(x(t_1)) & \dots & F_N(x(t_1)) \\ \vdots & \ddots & \vdots \\ F_1(x(t_{M-1})) & \dots & F_N(x(t_{M-1})) \end{bmatrix} \in \mathbb{R}^{M \times N} \tag{5}$$

This leads to n independent equations similar to (4):

$$\mathbf{y}_i = \mathbf{\Phi}\mathbf{w}_i + \mathbf{\Xi}_i, \ (i = 1, \dots, n). \tag{6}$$

where $\mathbf{w}_i = [w_{1i}, w_{2i}, \dots, w_{Ni}]^T \in \mathbb{R}^N$. We will introduce in section IV a method that allows us to reconstruct $\mathbf{w}_i$ from time-series observations of x (used to construct $\mathbf{\Phi}$) and $\mathbf{y}_i$.

B. Problem Transformation

Once the matrix $\mathbf{\Phi}$ 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 find the solution $\mathbf{w}_i$ of the linear regression problem defined in (6). Since there would be n independent linear regression problems, we can just consider one single problem and omit the subscripts i in (6) for simplicity of notation. We then write

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{\Xi}. \tag{7}$$

Typically the weighting vector $\mathbf{w}$ solution of (7) 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 vectors. On one hand, since the nonlinear form of the equation is typically unknown, there can potentially be a very large number of candidate functions. 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, BRN are typically sparse [9]. As a consequence, we typically have $N \gg M$ for $\mathbf{\Phi}_{M\times N}$ and $\mathbf{w}$ sparse.

The linear regression problem (7) can thus be defined as a compressive sensing, or sparse signal recovery problem [10], [11], with observation vector $\mathbf{y}_i$, known regressor matrix $\mathbf{\Phi}$, unknown coefficients $\mathbf{w}$, and additive noise Ξ. In sparse problems, the prior belief is that only a small fraction of the elements appearing in $\mathbf{w}$ are non-negligible. The general aim is to identify the smallest subset of columns of $\mathbf{\Phi}$, whose linear span contains the observations $\mathbf{y}$.

IV. COMPRESSIVE SENSING

A. Algorithm for Compressive Sensing

Since we want to get the sparsest solution of $\mathbf{w}$, we impose a penalty on the ℓ_0-norm of $\mathbf{w}$, $\|\mathbf{w}\|_0$, i.e., on the number of nonzero terms in $\mathbf{w}$. This leads to the regularisation regression problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\{\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \rho\,\|\mathbf{w}\|_0\}. \tag{8}$$

where ρ is a tradeoff parameter. Unfortunately, this optimisation problem is both numerically unstable and NP-complete; therefore some relaxations are typically used to recast this problem into another one for which efficient algorithmic solutions exist. The most common relaxation is to use the ℓ_1-norm instead of the ℓ_0-norm [10], so that the optimisation problem becomes

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}\{\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \rho\,\|\mathbf{w}\|_1\}. \tag{9}$$

The approach presented in eq. (9) is known as *Lasso*. A number of methods have been proposed to solve *Lasso* problems, including ℓ_1-minimisations (convex optimisation) [10] and greedy algorithms [12]. A sufficient condition for exact reconstruction with both of these algorithms is the so called *restricted isometry property* (RIP) [10]: A matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ 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 $(1-\delta)\,\|q\|_2^2 \leq \|\mathbf{\Phi}_I q\|_2^2 \leq (1+\delta)\,\|q\|_2^2$, where $\mathbf{\Phi}_i$ denotes the matrix formed by the columns of $\mathbf{\Phi}$ with indices in I. It was shown in [10], [12] that both ℓ_1-minimisations and greedy algorithms lead to exact reconstruction of K-sparse signals if the matrix $\mathbf{\Phi}$ satisfies the RIP with a constant parameter $0 \leq \delta \leq 1$. One major drawback is that RIP can be difficult to check. Another related and easier-to-check property is "coherence". The *coherence* of a matrix $\mathbf{\Phi}_i$ is defined as

$$\mu(\mathbf{\Phi}) = \max_{j<k} \frac{|\langle \Phi_j, \Phi_k\rangle|}{\|\Phi_j\|_2 \|\Phi_k\|_2}, \tag{10}$$

where Φ_j and Φ_k denote the j^{th} and k^{th} columns of $\mathbf{\Phi}$, respectively. It was shown that RIP guarantees *incoherence* of $\mathbf{\Phi}$, i.e. $\mu(\mathbf{\Phi}) = 0$. This means that ℓ_1 relaxations lead to exact reconstruction only when $\mathbf{\Phi}$ is orthogonal: columns of $\mathbf{\Phi}$ are strongly uncorrelated.

However, in real applications, correlation between the columns of $\mathbf{\Phi}$ is usually high ($\mu(\mathbf{\Phi})$ is close to 1). Since the dictionary matrix $\mathbf{\Phi}$ is basically composed of time series data, it is difficult to *a priori* guarantee low correlation. Sometimes $\mathbf{\Phi}$ even suffers rank deficiency. Consequently, given real data, the ℓ_0 regularisation regression problem in (8) can be hard to solve as an ℓ_1 relaxation problem (see (9)) using convex optimisation or greedy algorithms. A different approach thus needs to be considered. Thanks to recent results in machine learning, we propose hereafter a method intended to solve the compressive sensing problem defined in (8) in situations where ℓ_1 relaxations usually do not work. Our approach uses a Bayesian formulation [13]–[15].

B. Bayesian formulation of the CS problem

Bayesian modelling treats all unknowns as stochastic variables with certain probability distributions [16]. In the problem formulation $\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \Xi$, we assume that the stochastic variables in Ξ are independent and characterised by a Gaussian distribution with zero mean and variance σ^2. We further define the precision or inverse-variance as $\beta = 1/\sigma^2$. The data likelihood can then be shown to be

$$\begin{aligned} p(\mathbf{y}|\mathbf{w}) &= \mathcal{N}(\mathbf{y}|\mathbf{\Phi}\mathbf{w}, \beta^{-1}) \\ &= (2\pi\sigma^2)^{-\frac{M}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2\right) \end{aligned} \tag{11}$$

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

The sparseness of $\mathbf{w}$ can be imposed by using a prior of the form

$$p(\mathbf{w}) \propto \prod_j \exp(-|w_j|^p) = \exp(-\sum_j |w_j|^p). \tag{12}$$

where $\lambda_i > 0$, $0 < p \leq 1$. Combining $p(\mathbf{y}|\mathbf{w})$ and $p(\mathbf{w})$, we get the posterior distribution

$$p(\mathbf{w}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{w})p(\mathbf{w}) \tag{13}$$

Based on the above, we can formulate a *maximum a posteriori* (MAP) estimate for $\mathbf{w}$:

$$\begin{aligned} \mathbf{w}_{\text{MAP}} &= \arg\max_{\mathbf{w}} p(\mathbf{w}|\mathbf{y}) \\ &= \arg\min_{\mathbf{w}} -\log p(\mathbf{y}|\mathbf{w}) - \log p(\mathbf{w}) \\ &= \arg\min_{\mathbf{w}}\{\frac{\beta}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_j |w_j|^p\}. \end{aligned} \tag{14}$$

Specifically, when $p = 1$, (14) becomes equivalent to the ℓ_1-regularisation formulation in (9) which is convex; when $0 <$

$p < 1$, the cost function is not convex anymore and the MAP estimation has no guarantee to find the global minimum.

In the following section, we show that the use of hierarchical priors alleviates the afore mentioned problem. The CS problem can thus be converted into a linear-regression problem with a prior which is sparse. Given the sensing matrix $\mathbf{\Phi}$, we show in the next section how the sparse weights $\mathbf{w}$ and the inverse of noise variance β can be estimated using a sparse Bayesian learning approach.

V. RECONSTRUCTION VIA SPARSE BAYESIAN LEARNING

A. Specification of Hierarchical Priors

Instead of imposing a prior on $\mathbf{w}$, as defined in the last section, a SBL approach is adopted. Namely, we use hierarchical priors over the distribution of $\mathbf{w}$. The main advantage of such an approach is that it allows us to impose a Gaussian prior with zero-mean on each element $w_j \in \mathbf{w}$, i.e.,

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{N} \mathcal{N}(w_j|0, \alpha_j^{-1}). \tag{15}$$

In (15), $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^{1\times N}$ represents a vector of N independent hyperparameters, with α_j controlling the precision (or the inverse of the variance) of the prior imposed on $\mathbf{w}$. It is this form of prior that is eventually responsible for the sparsity properties of the model (see [13] for more details). It is common to place a Gamma prior on $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\alpha}|a, b) = \prod_{j=1}^{N} \Gamma(\alpha_j|a, b). \tag{16}$$

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, $a > 0$ is the shape parameter, and $b > 0$ 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 $\mathbf{w}$ is then evaluated as

$$p(\mathbf{w}|a, b) = \prod_{j=1}^{N} \int_0^\infty \mathcal{N}(w_j|0, \alpha_j^{-1})\Gamma(\alpha_j|a, b)d\alpha_j. \tag{17}$$

The density function $\Gamma(\alpha_j|a, b)$ is the conjugate prior for α_j when w_j plays the role of observed data and $\mathcal{N}(w_j|0, \alpha_j^{-1})$ is the associated likelihood function. Based on this, the integral $\int_0^\infty \mathcal{N}(w_j|0, \alpha_j^{-1})\Gamma(\alpha_j|a, b)d\alpha_j$ can be evaluated analytically. It can be shown that this integral corresponds to the Student's *t*-distribution which is strongly peaked around $w_j = 0$. Consequently, the prior in (17) is a sparseness prior for $\mathbf{w}$. Similarly, a Gamma prior is introduced on β

$$p(\beta|c, d) = \Gamma(\beta|c, d). \tag{18}$$

B. Bayesian Inference via Relevance Vector Machine

Given $\mathbf{y}$ and $\mathbf{\Phi}$ and assuming that the hyperparameters $\boldsymbol{\alpha}$ and β are known, the *posterior* distribution for $\mathbf{w}$ conditioned on the data can be obtained by combining the likelihood and prior with Bayes' rule. After some calculations, this yields:

$$p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}, \beta) = \frac{p(\mathbf{y}|\mathbf{w}, \beta)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{y}|\boldsymbol{\alpha}, \beta)} = \mathcal{N}(\mathbf{m}, \mathbf{\Sigma}). \tag{19}$$

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

$$\mathbf{m} = \beta\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{y} \tag{20}$$

$$\mathbf{\Sigma} = (\mathbf{A} + \beta\mathbf{\Phi}^T\mathbf{\Phi})^{-1} \tag{21}$$

where $\mathbf{A} = \mathrm{diag}(\alpha_j)$. In the context of Relevance Vector Machines (RVM), the associated "learning" problem becomes the search for the hyperparameters $\boldsymbol{\alpha}$ and β. RVM is a Bayesian sparse kernel method for regression and classification [13]. It shares a lot of characteristics with support vector machines [16]. In RVM, these hyperparameters are estimated from the data by maximising

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

This is known as the marginal likelihood. Its maximisation is known as evidence approximation maximisation [17].

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

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}, \beta) &= \log p(\mathbf{y}|\boldsymbol{\alpha}, \beta) \\
&= \log \int p(\mathbf{y}|\mathbf{w}, \beta)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \\
&= -\frac{1}{2}\left[M\log 2\pi + \log|\mathbf{C}| + \mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}\right]
\end{aligned} \tag{22}$$

where the $M \times M$ matrix $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T$. A type-2 maximum likelihood approximation employs the point estimates for $\boldsymbol{\alpha}$ and β to maximise $\mathcal{L}(\boldsymbol{\alpha}, \beta)$. By setting the required derivatives of the marginal likelihood to zero, we can obtain the following re-estimate equations [13], [17]

$$\alpha_j^{new} = \frac{\gamma_j}{m_j^2}, (j = 1, 2, \ldots, N), \tag{23}$$

where the posterior mean weight m_j is the jth element of $\mathbf{m}$ appearing in (20) and $\gamma_j \triangleq 1 - \alpha_j\mathbf{\Sigma}(jj)$, with $\mathbf{\Sigma}(jj)$ representing the jth diagonal element of the posterior weight covariance matrix $\mathbf{\Sigma}$ defined in (21). Estimation of γ_j and m_j can be efficiently done via the Expectation-Maximisation (EM) algorithm, an iterative procedure for maximum likelihood parameter estimation from data sets with missing or hidden variables. For a noise variance $\sigma^2 = 1/\beta$, the re-estimate equation is

$$(\beta^{new})^{-1} = \frac{\|\mathbf{y} - \mathbf{\Phi}\mathbf{m}\|_2^2}{M - \sum_j \gamma_j}. \tag{24}$$

Note that α_j^{new}, $j = 1, \ldots, N$ and β^{new} are functions of $\mathbf{m}$ and $\mathbf{\Sigma}$. Furthermore, $\mathbf{m}$ and $\mathbf{\Sigma}$ are a function of $\boldsymbol{\alpha}$ and β in equations (20)-(21). This suggests an iterative algorithm, which iterates between equations (20)-(21) and equations (23)-(24) until a convergence criterion has been satisfied. During this iteration process, many of the α_j tend to infinity when the corresponding w_j are very small or zero. Usually, only a few α_j are small. From (19), this implies that $p(w_j|\mathbf{y}, \alpha_j, \beta)$ becomes highly (in principle, infinitely) peaked at zero. This implies that we can be *a posteriori* 'certain' that w_j is zero. The corresponding basis functions in $\mathbf{\Phi}$ can thus be 'pruned', and sparsity of $\mathbf{w}$ can be achieved [13]. Computation of covariance in (21) must invert $N \times N$ matrices which is a $\mathcal{O}(N^3)$ procedure, thereby making EM algorithms relatively slow if a large number of basis functions is considered. Fortunately, a fast marginal likelihood maximisation was developed in [14], [15] by analysing the properties of the log marginal likelihood function in eq. (22). This enables a principled and efficient sequential addition and deletion of candidate basis function (columns of $\mathbf{\Phi}$) to monotonically maximise the log marginal likelihood.

C. Fast Marginal Likelihood Maximisation

To efficiently optimise (22) on one hyperparameter α_l with all other hyperparameters fixed, we can decompose (22) as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \beta) &= -\frac{1}{2}[M \log 2\pi + \log|\mathbf{C}_{-l}| + \mathbf{y}^T\mathbf{C}_{-l}^{-1}\mathbf{y} \\ &\quad - \log \alpha_l + \log(\alpha_l + s_l) - \frac{q_l^2}{\alpha_l + s_l}] \\ &= \mathcal{L}(\boldsymbol{\alpha}_{-l}, \beta) \\ &\quad + \frac{1}{2}\left[\log \alpha_l - \log(\alpha_l + s_l) + \frac{q_l^2}{\alpha_l + s_l}\right] \qquad (25)\end{aligned}$$

where $s_l = \phi_l^T \mathbf{C}_{-l}^{-1} \phi_l$, $q_l = \phi_l^T \mathbf{C}_{-l}^{-1} \mathbf{y}$, $\mathbf{C}_{-l} = \sigma^2 \mathbf{I} + \sum_{k \neq l} \alpha_l^{-1} \phi_k \phi_k^T$, and ϕ_l is the l^{th} column of $\mathbf{\Phi}$. Since $\mathcal{L}(\boldsymbol{\alpha}_{-l}, \beta)$ is independent of α_l, the maximum of (25) w.r.t. α_l is obtained at [15]: $\alpha_l = \begin{cases} \frac{s_l^2}{q_l^2 - s_l}, & \texttt{if} \quad q_l^2 > s_l, \\ \infty, & \texttt{if} \quad q_l^2 \leq s_l. \end{cases}$ Thus the marginal likelihood can be maximised w.r.t. one single hyperparameter at a time. This fast algorithm has a computational complexity of $\mathcal{O}(NM^2)$.

VI. IDENTIFICATION OF THE REPRESSILATOR

We consider here a classical dynamical system in systems/synthetic biology which we will use to illustrate the BRN reconstruction problem at hand. The repressilator is a synthetic oscillator network that was conceived and constructed by Elowitz and Leibler [18]. 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 is described as

$$\begin{aligned}\frac{dx_1}{dt} &= -\gamma_1 x_1 + \frac{\alpha_1}{1 + x_6^{n_1}} + \theta_1, \frac{dx_4}{dt} = -\gamma_4 x_4 + \beta_1 x_1, \\ \frac{dx_2}{dt} &= -\gamma_2 x_2 + \frac{\alpha_2}{1 + x_4^{n_2}} + \theta_2, \frac{dx_5}{dt} = -\gamma_5 x_5 + \beta_2 x_2, \\ \frac{dx_3}{dt} &= -\gamma_3 x_3 + \frac{\alpha_3}{1 + x_5^{n_3}} + \theta_3, \frac{dx_6}{dt} = -\gamma_6 x_6 + \beta_3 x_3.\end{aligned} \qquad (26)$$

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 (26) 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.

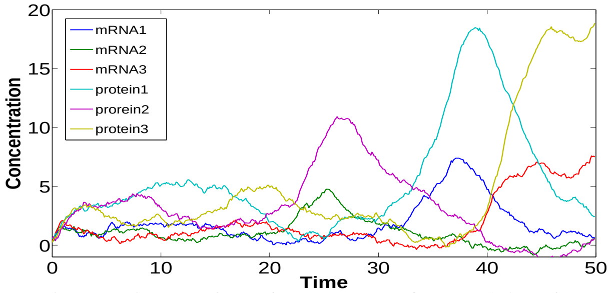

Fig. 1: Time series of $x_1, \ldots, x_6$ for model (26).

Take gene 1 for example. The hill coefficient n_1 will typically have a range from 1 to 6 due to biochemical constraints. The core question here is: how can we determine the nonlinear structure and kinetic parameters of the ODEs in (26)? Note that we do not assume *a priori* knowledge of the nonlinear functions forms, e.g., whether the degradation obeys first-order or enzymatic catalysed dynamics or whether the proteins are repressors or activators.

Next we show how the network reconstruction problem of the repressilator model in (26) can be formulated under the form presented in (6). Following the procedure in (2) and (3), we construct a matrix of candidate functions $\mathbf{\Phi}$ by selecting the most commonly used candidate basis functions used to model BRN. 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: $\text{hill}(t_k) \triangleq \left[\frac{1}{1+x_1^i(t_k)}, \ldots, \frac{1}{1+x_6^i(t_k)}, \frac{x_1^i(t_k)}{1+x_1^i(t_k)}, \ldots, \frac{x_6^i(t_k)}{1+x_6^i(t_k)}\right]_{1\times 12}$, where i represents the Hill coefficient. In what follows we consider that the Hill coefficient can take integers from 1 to 6. To also take into account basal transcription expression rates, we add to the last column of $\mathbf{\Phi}$ a unit vector. Since there are 6 state variables, we can construct the basis matrix $\mathbf{\Phi}$ appearing in (VI) with 6 (basis functions for linear terms) $+(6 * 12) = 72$ (basis functions for hill functions) $+1$ (basis function for basal expression) $= 79$ columns, e.g. $\mathbf{\Phi}(t_k) = [x_1(t_k), \ldots, x_6(t_k), \text{hill}_1(t_k), \ldots, \text{hill}_6(t_1), 1] \in \mathbb{R}^{6+72+1}$. Assuming $k = 0, \ldots, M-1$, we then construct $\mathbf{\Phi} = \left[\mathbf{\Phi}^T(t_0), \ldots, \mathbf{\Phi}^T(t_{M-1})\right]^T \in \mathbb{R}^{M\times 79}$.

Considering (6) with the basis function matrix $\mathbf{\Phi}$ mentioned above, the corresponding target weight matrix $\mathbf{w}$ should be:

$$\begin{bmatrix} -\gamma_1 & 0 & 0 & \beta_1 & 0 & 0 \\ 0 & -\gamma_2 & 0 & 0 & \beta_2 & 0 \\ 0 & 0 & -\gamma_3 & 0 & 0 & \beta_3 \\ 0 & 0 & 0 & -\gamma_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma_6 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ \alpha_1 & \alpha_2 & \alpha_3 & 0_{72\times1} & 0_{72\times1} & 0_{72\times1} \\ 0_{54\times1} & 0_{56\times1} & 0_{55\times1} & & & \\ \theta_1 & \theta_2 & \theta_3 & 0 & 0 & 0 \end{bmatrix}.$$

As an illustration, we define $\mathbf{w}_{\texttt{true}}$ as with chosen values:

$$\begin{bmatrix} -0.3 & 0 & 0 & 1.4 & 0 & 0 \\ 0 & -0.4 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & -0.5 & 0 & 0 & 1.6 \\ 0 & 0 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.6 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ 4 & 3 & 5 & 0_{72\times1} & 0_{72\times1} & 0_{72\times1} \\ 0_{54\times1} & 0_{56\times1} & 0_{55\times1} & & & \\ 0.02 & 0.02 & 0.01 & 0 & 0 & 0 \end{bmatrix}.$$

Let $t_T = 50$, sampling interval $t_{k+1} - t_k = 1$, and $M = 50$ points are collected. The noise variance is fixed at 10^{-2}. We run the algorithm with 100 runs. Among the sparsest solutions, we select the one with minimal $\|\mathbf{w}_{\texttt{estimate}} - \mathbf{w}_{\texttt{true}}\|$. The chosen solution $\mathbf{w}_{\texttt{estimate}}$ is

$$\begin{bmatrix} 0.300 & 0 & 0 & 1.401 & 0 & 0 \\ 0 & -0.398 & 0 & 0 & 1.502 & 0 \\ 0 & 0 & -0.499 & 0 & 0 & 1.600 \\ 0 & 0 & 0 & -0.200 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.400 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.600 \\ 0_{17\times1} & 0_{15\times1} & 0_{16\times1} & & & \\ 3.999 & 3.002 & 5.001 & 0_{72\times1} & 0_{72\times1} & 0_{72\times1} \\ 0_{54\times1} & 0_{56\times1} & 0_{55\times1} & & & \\ 0.019 & 0.019 & 0.0093 & 0 & 0 & 0 \end{bmatrix}.$$

The algorithm is implemented in MATLAB R2012a. The calculation is run on a standard laptop computer (Intel Core Duo i5-5250 2.50GHz with 8GB RAM). The computation time for each run is less than $\mathbf{0.1}$ second.

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 BRN) are sought after. 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.

We have so far assumed that the system is fully observable. However, in reality, measurement observations will typically be partial [19] (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 and Dr Jorge Gonçalves 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 (project EP/G036004/1) at Imperial College. The authors would like to thank Dr Wei Dai at Imperial College for the constructive discussions that lead to this paper.

REFERENCES

[1] 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, no. 12, pp. 1877–1886, 2004.

[2] E. August and A. Papachristodoulou, "Efficient, sparse biological network determination," *BMC Systems Biology*, vol. 3, no. 1, p. 25, 2009.

[3] M. Zavlanos, A. Julius, S. Boyd, and G. Pappas, "Inferring stable genetic networks from steady-state data," *Automatica*, 2011.

[4] Y. Yuan, "Decentralised network prediction and reconstruction algorithms," Ph.D. dissertation, Cambridge University, 2012.

[5] 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, no. 13, p. 7702, 2003.

[6] L. Ljung, H. Hjalmarsson, and H. Ohlsson, "Four encounters with system identification," *European Journal of Control*, vol. 17, no. 5, p. 449, 2011.

[7] C. Rick, "Numerical differentiation of noisy, nonsmooth data," *ISRN Applied Mathematics*, vol. 2011, 2011.

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

[9] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[10] E. Candès and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[11] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[12] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2230–2249, 2009.

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

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

[15] 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, no. 3. Jan, 2003.

[16] C. Bishop, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4.

[17] D. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[18] M. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–8, 2000.

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