Identification of nonlinear sparse networks using sparse Bayesian learning

Junyang Jin, Ye Yuan†, Wei Pan, Claire Tomlin, Alex A.Webb, Jorge Gonçalves

***Abstract*— This paper considers a parametric approach to infer sparse networks described by nonlinear ARX models, with linear ARX treated as a special case. The proposed method infers both the Boolean structure and the internal dynamics of the network. It considers classes of nonlinear systems that can be written as weighted (unknown) sums of nonlinear functions chosen from a fixed (and potentially large) basis dictionary. Due to the sparse topology, coefficients of most groups are zero. Besides, only a few nonlinear terms in nonzero groups contribute to the internal dynamics. Therefore, the identification problem should estimate both group- and element-sparse parameter vectors. The proposed method combines Sparse Bayesian Learning (SBL) and Group Sparse Bayesian Learning (GSBL) to impose both kinds of sparsity. Simulations indicate that our method outperforms SBL and GSBL when these are applied alone. A linear ring structure network also illustrates that the proposed method has improved performance compared to the kernel approach.**

I. INTRODUCTION

When designing feedback controllers, it is typically enough to learn the input-output dynamics of the system, independently of its internal complexity. Hence, most of the work on system identification focuses on modeling input-output dynamics without exploring the topology, i.e., the interactions between state-variables. In many applications, however, information about the topology is critical. We may require an understanding of the internal dynamics, to provide a road-map and guidance for feedback control, or to find source of faults. Examples range from biomedicine to autonomous underwater vehicle (AUVs), power networks and communication networks.

Sparsity is an inherent property of many important networks. In biology, most molecules bind with to a small number of other molecules. In AUVs, communication can be constrained to neighbours to minimise energy consumption. Elements of power and communication networks are typically connected to a small number of other elements. Hence, sparsity can be used as a constraint to model networks and compensate for the sometimes low number of samples and high amount of noise.

Junyang Jin and Alex Webb are with Circadian Signal Transduction Group, Department of Plant Sciences, University of Cambridge. Ye Yuan is with School of Automation, Huazhong University of Science and Technology. Wei Pan is with Cardwell Investment Tech and Imperial College London. Claire J. Tomlin is with the Department of Electrical Engineering and Computer Sciences, UC Berkeley. Jorge Gonçalves is with the Department of Engineering, University of Cambridge and the Luxembourg Centre for Systems Biomedicine. †For correspondence, yye@hust.edu.cn.

Standard system identification methods, such as the prediction error method (PEM) or Maximum-likelihood (ML), are widely used to explore input-output dynamics of systems [1]. However, these methods alone do not capture sparse topology. For noisy MIMO systems, assuming no prior knowledge of the topology, PEM generates full transfer matrices even if the ground truths are sparse [2]. Hence, methods must penalise model complexity to favour sparsity.

MAP (Type I method) including LASSO, Tikhonov regularisation, FOCUSS and sparse group LASSO (SGL) [3], [4] are all methods that penalise model complexity. For example, algorithms were used to infer the topology of a linear MIMO system from steady-state data [5], [6]. Similar work inferred sparse multivariable ARX models with known polynomial order using Block Orthogonal Matching Pursuit (BOMP) [7]. While these approaches effectively reduce over-fitting, the tuning variable which controls the trade-off between data-fitting and model complexity (sparsity) must be *a priori* chosen or evaluated independently (using methods such as cross-validation). This increases the computation burden and causes information waste.

There are alternative methods that do not require extra efforts to evaluate tuning variables such as Sparse Bayesian Learning (Type II method) and kernel methods. The estimation of tuning variables (namely hyperparameter) is incorporated into the identification process following Bayesian perspective. SBL is a well-known technique in machine learning that applies a nonconvex penalty to approximate ℓ_0 norm [8]–[11]. It was applied to identify nonlinear systems by selecting nonlinear functions from a predefined dictionary [12], [13]. Nonlinear model structure is captured by either element SBL or GSBL, depending on the type of available data. However, the topology is not carefully considered. The kernel method is a non-parametric approach, introduced to estimate impulse responses of SISO systems [14]. Later it was combined with GSBL to infer linear sparse networks [2], [15], [16].

This paper considers the parametric identification of a sparse network described by a nonlinear multivariable ARX model. The objective is two-fold: to infer the network topology while also achieving accurate estimation of model parameters. The identification problem is formulated as to estimating a target vector consisting of model parameters that is both group sparse (due to the network topology) and element sparse (in terms of candidate nonlinear functions). Our proposed scheme combines SBL and GSBL to simultaneously impose those two kinds of sparsity. Simulations on a

three-gene repressilator model show that our method is better than SBL or GSBL applied alone. Inference of ring structure networks further highlights the strength of our method to deal with extremely sparse networks.

The paper is organized as follows. Section II introduces the nonlinear multivariable ARX model. Section III formulates the network reconstruction problem. Section IV promotes a sparse prior and applies EM algorithm to solve the resulting optimization problem. Section V compares the method with other approaches via Monte Carlo simulation. Finally, Section VI concludes and discusses further development in this field.

Notation: The notation in this paper is standard. I denotes the identity matrix. If $L \in R^{n\times n}$, diag$\{L\}$ denotes a vector which consists of diagonal elements of the matrix L. If $l \in R^n$, diag$\{l\}$ denotes a diagonal matrix with its diagonal elements to be the vector l. For a series of matrices $\{A_1, ..., A_n\}$, blkdiag$\{A_1, ..., A_n\}$ presents a block diagonal matrix. vec$\{x_1, .., x_n\} = [x_1, ..., x_n]'$ means to vectorise elements $\{x_1, .., x_n\}$. A vector $y(t_1 : t_2)$ denotes a row vector $\begin{bmatrix} y(t_1) & y(t_1+1) & \cdots & y(t_2) \end{bmatrix}$. A_{ij} denotes the element at ith row and jth columm of the matrix A.

II. MODEL FORMULATION

The sparse network is described by a nonlinear multivariable ARX model: $A(q^{-1})Y(t) = B(q^{-1})U(t)+F(t)+E(t)$, where

$$\begin{aligned} A(q^{-1}) &= I + \hat{A}_1 q^{-1} + ... + \hat{A}_{n_a} q^{-n_a}, \\ B(q^{-1}) &= \hat{B}_1 q^{-1} + ... + \hat{B}_{n_b} q^{-n_b}, \end{aligned} \tag{1}$$

q^{-1} is the time shift operator. $Y(t) \in R^p$ are the nodes of the network, $U(t) \in R^m$ denotes input, and $E(t) \in R^p$ is i.i.d white Gaussian noise. $\hat{A}_i \in R^{p\times p}$ and $\hat{B}_i \in R^{p\times m}$ are matrices. $A(q^{-1})$ is a polynomial matrix showing the connectivity of each node to the others including self-loops. Similarly, $B(q^{-1})$ is a polynomial matrix relating the input to the nodes. $F(t)$ is a vector of nonlinear functions depending on the past values of nodes and input. Each element of $F(t)$ is the linear combination of basis functions. The Boolean structure of the network is reflected by the nonzero elements in $A(q^{-1})$, $B(q^{-1})$, and nonlinear terms of $F(t)$ whereas the system dynamics is dominated by the elements in these matrices.

If $F(t)$ is set to 0, the model becomes a linear ARX. The multivariable ARX model defines a unique Input-Output map:

$$Y(t) = G(z^{-1})U(t) + H(z^{-1})E(t), \tag{2}$$

where

$$\begin{aligned} G(z^{-1}) &= A^{-1}(z^{-1})B(z^{-1}) \\ H(z^{-1}) &= A^{-1}(z^{-1}). \end{aligned} \tag{3}$$

III. RECONSTRUCTION PROBLEM FORMULATION

We parameterize each node of a nonlinear multivariable ARX model in the similar form:

$$\begin{aligned} &y_i(t) = -A_{i1}(q^{-1})y_1(t) - \ldots - A_{ip}(q^{-1})y_p(t)+ \\ &[1 - A_{ii}(q^{-1})]y_i(t) + ... + B_{im}(q^{-1})u_m(t) + F_i(t) + E_i(t). \end{aligned} \tag{4}$$

$y_j(t)$ denotes jth node, $u(t) \in R^m$ input, $E_i(t)$ i.i.d Gaussian noise and:

$$\begin{aligned} A_{ii}(q^{-1}) &= a_1^{ii}q^{-k} + a_2^{ii}q^{-k+1} + \ldots + a_k^{ii}q^{-1} + 1 \\ A_{ij}(q^{-1}) &= a_1^{ij}q^{-k} + \ldots + a_{(k-1)}^{ij}q^{-2} + a_k^{ij}q^{-1} \\ B_{ij}(q^{-1}) &= b_1^{ij}q^{-k} + \ldots + b_{(k-1)}^{ij}q^{-2} + b_k^{ij}q^{-1} \\ F_i(t) &= \sum_{j=1}^{p}\sum_{q=1}^{l} \theta_q^{ij} f_q^{ij}(t) \\ f_q^{ij}(t) &= g_q^{ij}[y_j(t-k:t-1), u(t-k:t-1)] \end{aligned} \tag{5}$$

where a and b denote parameters in polynomial matrices $A(q^{-1})$ and $B(q^{-1})$ respectively, q^{-1} the time shift operator, superscript ij the polynomial of ijth element of matrices and subscript i the index of the ith coefficient of the polynomial. The order k is the maximum time delay. $F_i(\cdot)$ is the linear combination of nonlinear basis functions $g(\cdot)$ which depends on the past evolution. The time delay of nonlinear terms is flexible as long as it is smaller than k. For convenience, the time delay in all terms is unified to k. The vector θ^i is divided into p groups corresponding to the number of nodes with l elements in each group. θ_q^{ij} denotes the qth element of the jth group of θ^i.

Assume time-series data from discrete time indices 1 to t for each node and input is available. For the ith node, we define the following matrices and vectors:

$$y = \begin{bmatrix} y_i(t) \\ \vdots \\ y_i(k+1) \end{bmatrix}, w = \begin{bmatrix} w_1 \\ \hline \vdots \\ \hline w_{p+m} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} \Phi_1 \mid ... \mid \Phi_{p+m} \end{bmatrix}$$

$$\lambda = E\{E_i(t)^2\}, E\{E_i(t)\} = 0$$

$$w_j = \begin{cases} \begin{bmatrix} a_1^{ij} & \ldots & a_k^{ij} \mid \theta_1^{ij} & \ldots & \theta_l^{ij} \end{bmatrix}^T, & j \le p \\ \begin{bmatrix} b_1^{i(j-p)} & \ldots & b_k^{i(j-p)} \end{bmatrix}^T, & p < j \le p+m \end{cases}$$

If $j \le p$

$$\Phi_j = \left[\begin{array}{ccc|ccc} -y_j(t-k:t-1) & & & f_1^{ij}(t) & ... & f_l^{ij}(t) \\ \vdots & & & & \vdots & \\ -y_j(1:k) & & & f_1^{ij}(k+1) & ... & f_l^{ij}(k+1) \end{array}\right]$$

elseif $p < j \le p+m$

$$\Phi_j = \begin{bmatrix} u_{j-p}(t-k:t-1) \\ \vdots \\ u_{j-p}(1:k) \end{bmatrix}$$

(6)

The likelihood function based on Bayes' rule is thus:

$$p(y|w,\lambda) = \frac{1}{(2\pi\lambda)^{(t-k)/2}} \exp\{-\frac{1}{2\lambda}\|y-\Phi w\|_2^2\}. \quad (7)$$

By maximizing its logarithm with respect to w, we end with the PEM (ML) method. In practice, given limited data, PEM may suffer from over-fitting and also generate a fully-connected network even if the true one is sparse. Therefore, penalties for both network topology and model complexity are essential. Referring to the parameterizations in (6), a sparse network can be interpreted as group sparse w, whereas sparsity within each group indicates that only few nonlinear terms out of a dictionary and reduced order polynomial are relevant to the dynamics of the network. For instance, a group of nonlinear terms are added to a gene regulatory network to describe the potential transcription activity of a transcriptional factor associated to a specific node. The group sparsity determines if this node regulates the target. Besides, only a specific type of the hill functions in this group is appropriate depending on whether such a transcription is repressive or active. We resort to Sparse Bayesian Learning to promote these two kinds of sparsity.

IV. INDUCING SPARSITY VIA SPARSE BAYESIAN LEARNING

A. Sparsity inducing priors

Full Bayesian treatment requires introducing a prior distribution for w to reflect its main features. Priors like Generalized Gaussian, Student's t and Logistic are normally used to induce sparsity of the parameter [11]. However, estimating w using its posterior mean is intractable in this case because the posterior distribution $p(w|y)$ is non-Gaussian and not analytical. Sparse Bayesian learning approximates $p(w|y)$ with a Gaussian distribution so that $E(w|y)$ can be easily calculated. A sparse prior is first presented in a variational form which yields a lower bound for that prior [17], [18]. The property of the lower bound is controlled by its hyperparameters. A designed criterion is then applied to find the most proper hyperparameters.

As discussed, the parameter vector w is both element sparse and group sparse. There are priors able to induce either of these two types of sparsity. We use these priors to construct a novel one that can impose both of them.

Sparse priors able to induce element sparsity to $w \in R^{k(p+m)+lp}$ can be expressed in the variational from as [11], [12]:

$$\begin{aligned} p(w) &= \prod_{i=1}^{p+m} p(w_i) = \max_{\beta\geq 0} \mathcal{N}(w|0,B)\varphi^{\beta}(\beta) \\ p(w_i) &= \prod_{j} p(w_{ij}) = \max_{\beta_i\geq 0} \mathcal{N}(w_i|0,B_i)\varphi_i^{\beta}(\beta_i) \\ p(w_{ij}) &= \max_{\beta_{ij}\geq 0} \mathcal{N}(w_{ij}|0,\beta_{ij})\varphi_{ij}^{\beta}(\beta_{ij}) \end{aligned} \quad (8)$$

where subscript i denotes ith group in a vector and j the jth element in that group. $\beta = \text{vec}\{\beta_1,...,\beta_{p+m}\}$ is a vector of hyperparameters which controls element sparsity of the vector w and $\beta_i = \text{vec}\{\beta_{i1},..,\beta_{i(k+l)}\}$ if $i \leq p$ or $\text{vec}\{\beta_{i1},..,\beta_{ik}\}$ if $p < i \leq p+m$. B is the covariance matrix of Gaussian distribution and parameterized by vector β as $B = \text{diag}\{\beta\}$ and $B_i = \text{diag}\{\beta_i\}$. $\mathcal{N}(w|\mu,\Sigma)$ denotes the Gaussian distribution of w with mean μ and covariance Σ. $\varphi_{ij}^{\beta}(\cdot)$ is a positive function which depends on the prior $p(w)$.

To impose group sparsity, the hyperparameters of each group are unified so that elements in a group share the same sparse profile [13], [19]:

$$\begin{aligned} p(w) &= \prod_{i=1}^{p+m} p(w_i) = \max_{\gamma\geq 0} \mathcal{N}(w|0,\Gamma)\varphi^{\gamma}(\gamma) \\ p(w_i) &= \max_{\gamma_i\geq 0} \mathcal{N}(w_i|0,\gamma_i I)\varphi_i^{\gamma}(\gamma_i) \end{aligned} \quad (9)$$

where γ is a vector of hyperparameters which controls group sparsity of w and $\Gamma = \text{blkdiag}\{\gamma_1 I,..,\gamma_{p+m} I\}$. $\varphi^{\gamma}(\cdot)$ is a positive function.

Remark 1: : In practice, if the model is linear ARX, the group of w that presents auto-regression can be excluded from the group prior to improve the estimation accuracy.

According to the construction of element and group sparse priors, neither of them is suitable to impose both element and group sparsity. The hyperparameters in (8) are independent so that the resulting prior is too flexible to impose group sparsity. The prior in (9) disables the element sparsity within each group thus too rigid. To promote both element and group sparsity, we combine (8) and (9) to get a new distribution:

$$p(w) = C \max_{\gamma\geq 0,\beta\geq 0} \mathcal{N}(w|0,B)\mathcal{N}(w|0,\Gamma)\varphi^{\beta}(\beta)\varphi^{\gamma}(\gamma) \quad (10)$$

where C is the normalization constant independent on hyperparameters γ and β which can be absorbed by positive functions $\varphi^{\beta}(\beta)$ or $\varphi^{\gamma}(\gamma)$.

Based on (10), we deduce an improper prior as the lower bound of the original one as:

$$\begin{aligned} \hat{p}(w) &= \mathcal{N}(w|0,B)\mathcal{N}(w|0,\Gamma)\varphi^{\beta}(\beta)\varphi^{\gamma}(\gamma) \\ &\leq p(w). \end{aligned} \quad (11)$$

The prior in (11) shows that two types of sparsity are controlled by two series of hyperparameters, β and γ respectively. As $\gamma_i \to 0$, the ith group of w is enforced to 0 regardless of β_i. That means the group sparsity can be determined from the hyperparameter space of dimension $p+m$ instead of $k(p+m)+lp$ if only element sparse priors are applied. Similarly, within a non-zero group ($\gamma_i \neq 0$), w_{ij} is driven to 0 if $\beta_{ij} \to 0$. As a result, for a target vector consisting of q non-zero groups out of p in total and k elements in each group, its sparse profile is determined from a $\{R^p\} \times \{R^{qk}\}$ hyperparameter space compared to $\{R^{(p-q)k}\} \times \{R^{qk}\}$ without group sparse priors. The values of these hyperparameters are unknown which remain to be estimated from the data.

Remark 2: : The conventional way to promote both element and group sparsity is to use hierarchical Bayesian. Two hyperparameters, β and γ are used for element and group sparsity respectively. The prior of w is parameterised by the hyperparameter β, while another prior conditioned on γ is introduced to β. As a result, the hyperparameter γ controls the group sparsity of w indrectly via β. However, the hyperparameter which is deeper in hierarchy has less impact on the inference procedure [20]. This means that the resulting penalty is weak in imposing group sparsity to w. As such, multiplying two priors makes sense since both hyperparameters influence w directly.

B. Type II Maximisation

Although the prior $\hat{p}(w)$ is improper, we can still get a normalized posterior distribution of w as:

$$\hat{p}(w|y) = \frac{p(y|w)\hat{p}(w)}{\int p(y|w)\hat{p}(w)dw}. \tag{12}$$

Clearly, $\hat{p}(w|y)$ is a Gaussian distribution:

$$\hat{p}(w|y) = \mathcal{N}(w|\mu, \Sigma), \tag{13}$$

where

$$\begin{aligned} \Sigma &= \left[(\Gamma^{-1} + B^{-1}) + \lambda^{-1}\Phi^T\Phi\right]^{-1} \\ \mu &= \lambda^{-1}\Sigma\Phi^T y \end{aligned} \tag{14}$$

It is clear that the sparsity of the estimated w from $E(w|y) = \mu$ depends on hyperparameters β and γ. Since they are unknown, SBL introduces Bayesian approximation to estimate their optimal values.

In what follows, vectors $\text{diag}\{\Sigma\}$ and μ are partioned into groups in the form of the hyperparameter β. To approximate the real $p(w|y)$, we minimize the misaligned mass between $p(w)$ and $\hat{p}(w)$ weighted by the marginal likelihood $p(y|w)$ which is also called evidence maximization or Type II maximization [11], [12], [21]. It is equivalent to estimating hyperparameters using the maximum likelihood method:

$$\begin{aligned} (\gamma^*, \beta^*, \lambda^*) &= \arg\min_{\beta,\gamma,\lambda\geq 0} \int p(y|w)|p(w) - \hat{p}(w)|dw \\ &= \arg\min_{\beta,\gamma,\lambda\geq 0} -2\log \int p(y|w)\hat{p}(w)dw \\ &= \arg\min_{\beta,\gamma,\lambda\geq 0} -2\log\ \hat{p}(y|\beta,\gamma,\lambda). \end{aligned} \tag{15}$$

Remark 3: It should be noticed that not all the sparse priors can lead to a sparse solution to (15) under the framework of Bayesian approximation. That is, the selection of the functions $\varphi^\beta(\cdot)$ and $\varphi^\gamma(\cdot)$ influences the sparsity of the final result. It is shown that one reasonable choice is that $-\log\varphi(\cdot)$ is concave and nondecreasing [11]. In this paper, we set $\varphi(\cdot)$ as a constant. Therefore, it can be ignored in the following discussion.

C. EM Algorithm to solve Type II maximisation

EM method is a traditional technique to solve (15). It belongs to the class of majorization-minimization (MM) method and is a special case of DCA (Difference of Convex functions Algorithm). To maximise a likehood function, $L(\theta) = \log p(y|\theta)$, EM implements Expectation (E step) and Maximisation (M step) iteratively. In the E step, the function, $Q(\theta, \theta^n) = E_{x|y,\theta^n}[\log p(y, x|\theta)] = \int \log p(y, x|\theta)p(x|y, \theta^n)dx$ is calculated where x is the unobservable latent random variable. In the M step, the optimisation problem, $\theta^{n+1} = \arg\max Q(\theta, \theta^n)$ is solved [21], [22]. The generated sequence, $\{\theta^n\}$ leads to the increased likelihood function ($L(\theta^n) < L(\theta^{n+1})$). In our case, we regard w as the latent variable. Following the standard procedure of EM method, the algorithm is described in Algorithm 1. To solve the optimisation problem (15) with the matrix $\Phi \in \mathbb{R}^{N\times M}$ and $N \ll M$, the cost of EM method is $O(MN^2)$ in each iteration.

Algorithm 1 Solve (15) using EM method

1: Initialize β^0, γ^0, λ^0
2: **for** $n = 1$: Max **do**
3: E step: Formulate $p(w|y, \beta^n, \gamma^n, \lambda^n)$ according to (13) and (14)
4: M step: Formulate the optimisation problem and update solutions as:

$$\begin{aligned} &[\beta^{n+1}, \gamma^{n+1}, \lambda^{n+1}] \\ &= \arg\min E_{w|\beta^n,\gamma^n,\lambda^n}\{\ln p(y, w|\beta, \gamma, \lambda)\} \end{aligned} \tag{16}$$

$$\begin{aligned} \gamma_i^{n+1} &= \begin{cases} \frac{1}{k+l}\sum_{j=1}^{k+l} \text{diag}\{\Sigma^n\}_{ij} + (\mu_{ij}^n)^2, i \leq p \\ \frac{1}{k}\sum_{j=1}^{k} \text{diag}\{\Sigma^n\}_{ij} + (\mu_{ij}^n)^2, p < i \leq p+m \end{cases} \\ \beta_{ij}^{n+1} &= \text{diag}\{\Sigma^n\}_{ij} + (\mu_{ij}^n)^2 \\ \lambda^{n+1} &= \frac{\|y - \Phi\mu^n\|_2^2 + \lambda^n \sum_{i=1}^{p+m}\sum_j 1 - \tau_{ij}\text{diag}\{\Sigma^n\}_{ij}}{N} \end{aligned} \tag{17}$$

where

$$\tau_{ij} = (\beta_{ij}^n)^{-1} + (\gamma_i^n)^{-1}, N = (k+l)p + km. \tag{18}$$

5: **if** some stopping criteria is satisfied **then**
6: Break;
7: **end if**
8: **end for**

V. SIMULATION

We present two simulations to illustrate our method. The first example is a three-gene repressilator network discussed in [12]. In the second example, linear ARX models are studied.

The topology inferences are evaluated using the average of True Positive Rate (TPR), the average of the Precision (Prec), and the percentage of successful inference (100% TPR and 100% Prec) among all runs. TPR reveals the percentage of how many true links of the ground truth networks are

identified. Prec indicates the confidence of estimation, which equals $TP/(TP+FP)$, where TP is the number of true links correctly identified and FP is the number of those incorrectly identified. For example, if Prec is 50%, it means that half of the links in the estimated network are wrong.

To evaluate the accuracy of estimated parameters, we calculate the normalised root mean square error (NRMSE) as:

$$\begin{aligned} NRMSE &= \frac{1}{\sqrt{N}\hat{x}}\|x_{est}-x_{true}\|_2 \\ \hat{x} &= \frac{1}{N}\|x_{true}\|_1. \end{aligned} \tag{19}$$

A. Gene regulatory network

The repressilator model describes the transcription and translation activities among three genes and proteins. Hill functions are used to represent dynamics of transcription while degradation and translation are described by linear terms. The model is given below:

$$\begin{aligned} x_1(k+1) &= (1-\delta_1)x_1(k) + \frac{\alpha_1}{1+x_6^{n_1}(k)} + u(k) + e_1(k) \\ x_2(k+1) &= (1-\delta_2)x_2(k) + \frac{\alpha_2}{1+x_4^{n_2}(k)} + e_2(k) \\ x_3(k+1) &= (1-\delta_3)x_3(k) + \frac{\alpha_3}{1+x_5^{n_3}(k)} + e_3(k) \\ x_4(k+1) &= (1-\delta_4)x_4(k) + \beta_1 x_1(k) + e_4(k) \\ x_5(k+1) &= (1-\delta_5)x_5(k) + \beta_2 x_2(k) + e_5(k) \\ x_6(k+1) &= (1-\delta_6)x_6(k) + \beta_3 x_3(k) + e_6(k), \end{aligned} \tag{20}$$

where

$$\begin{aligned} &\delta_1 = 0.3, \delta_2 = 0.4, \delta_3 = 0.5, \delta_4 = 0.2, \delta_5 = 0.4, \delta_6 = 0.6 \\ &\alpha_1 = 4, \alpha_2 = 3, \alpha_3 = 5, \beta_1 = 1.4, \beta_2 = 1.5, \beta_3 = 1.6 \\ &n_1 = 1, n_2 = 2, n_3 = 2. \end{aligned} \tag{21}$$

Variables x_1, x_2, x_3 denote the concentration of mRNAs of three genes whereas x_4, x_5, x_6 represent proteins. e denotes i.i.d Gaussian noise. u presents the stimuli into the network (known) and was set to be a step function with amplitude 0.01. Parameters of the model correspond to the rate of biochemical reactions. The nonlinear terms in the model are Hill functions describing repressive transcriptional activity. The model was simulated with different noise variance from time indices 1 to 50. Full state measurements were used for identification.

The objectives are to infer the topology of the network, estimate model parameters and recognize whether the gene regulations are repressive or active. Assuming no prior knowledge of the network, we built a dictionary of candidate functions including linear functions and Hill functions with the Hill coefficient from 1 to 4 in both repression and activation forms. For the ith node (we do not know whether it is mRNA or protein), there are 9 relevant functions:

$$\left[x_i, \frac{x_i}{1+x_i}, \frac{1}{1+x_i}, \frac{x_i^2}{1+x_i^2}, \frac{1}{1+x_i^2}, ..., \frac{x_i^4}{1+x_i^4}, \frac{1}{1+x_i^4}\right]. \tag{22}$$

Therefore, the target vector w for each node is of dimension $9\times 6+1 = 55$ (6 groups with 9 elements in each group plus an extra term of the input).

Successful identification should not only infer the correct network topology but also indicate the type of the gene regulation by selecting the proper Hill functions.

Table I compares the inferred Boolean structure of the repressilator network. With no process noise, our method and SBL achieve perfect inference ($TPR = 100\%$, $Prec = 100\%$). With relatively small noise variance ($1e-3$), while SBL and GSB inferred all the true linkgs, their Prec decreases to 53% and 45% respectively meaning it becomes difficult to tell which links in their inferred networks are true. In the contrast, our method still retains high Prec (81%). As the noise variance increases to $1e{-}1$, Prec of SBL and GSBL further drops to below 50%. Although our method missed some true links, the confidence of estimation is much higher than the other two methods ($Prec = 76\%$). On average, our method identified 6.6 links in total, among which only 1.6 links are wrong.

Table II shows that with no process noise, the estimation error of our method and SBL is negligible but the estimation accuracy of our method is more robust to the process noise.

B. Linear ARX models

Data was simulated from stable and sparse linear ARX models ($\theta = 0$) with 10 nodes. Our approach was compared with other methods: kernel method in [2], GSBL and SBL. For the kernel approach, the stable spline kernel was used to estimate the finite impulse responses of ARX models. Hyperparameters of kernel functions were estimated using Type II maximisation. Note that there are no obstacles preventing the application of our method to unstable networks since no constraint is required on the system property.

The simulation generated 100 random networks, from a fixed topology: a ring network as in Figure 1. Internal dynamics (polynomial matrices) up to 5th order were generated randomly. There is only one input applied to node 1. Both the exciting input (known) and process noise (unknown) were independent Gaussian. The ratio of input and noise variance was $SNR = 10\log_{10}(\sigma_u^2/\sigma_e^2) = 20$dB. The upper bound of the polynomial order (k in (5)) was set to 8.

For identification, the simulation collected 65 data points for each node and input. This class of networks is very sparse and contains a feedback loop.

The inference of ring networks further highlights the significance of Prec. The ring network contains only 10 links (of a total of 90 possible links). Hence, a high TPR is only meaningful if Prec is also high, or otherwise there is a low probability to choose a true link from all of those inferred. Therefore, the main task of inference in this case is to achieve

TABLE I: Inference results for the Repressilator network.

	No noise			1e-3 Var			1e-1 Var		
	Prec	TPR	Success	Prec	TPR	Success	Prec	TPR	Success
Our	100%	100%	100%	81%	98%	12%	76%	84%	8%
SBL	100%	100%	100%	53%	100%	0	36%	100%	0
GSBL	75%	100%	0	45%	100%	0	36%	100%	0

TABLE II: NRMSE of the Repressilator network.

	No noise	1e-3 Var	1e-1 Var
Our	1e-3	0.94	3.90
SBL	1e-3	1.28	4.07
GSBL	3.4	6.89	6.16

TABLE III: Inference results for ring networks using different methods.

	Prec	TPR	Succ	NRMSE
Our	93.2%	92.4%	49%	1.48
SBL	11.2%	100%	0	9.04
GSBL	13.2%	97.9%	0	7.99
Kernel	46.4%	95.2%	0	2.62

both high TRP and Prec. The results in Table III confirm this consideration. All the methods attain very high TPR ($> 90\%$). While our method has the lowest TPR (92.4%), it has by far the largest Prec (93.2%), meaning that on average 9.2 links are correctly inferred (out of 10) in a total of 9.9 links estimated (out of 90). The Prec of SBL and GSBL is extremely low, meaning that these methods estimated almost all 90 possible links. Even for the kernel method, its Prec is below 50% meaning that, on average, there are 20.5 links estimated but only 9.5 of them are correct. Finally, our method perfectly estimated (100% TPR and 100% Prec) 49% of all networks and had the lowest estimation error of all methods.

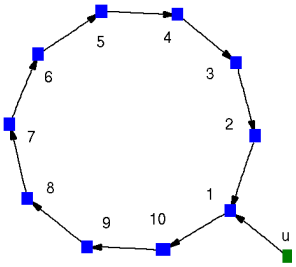

Fig. 1: A ring network.

VI. CONCLUSION AND DISCUSSION

This paper combines SBL and GSBL to identify nonlinear multivariable ARX models given measured time series data. No prior knowledge of the system is needed besides sparse topology. Motivated by simulation examples, the topology and model parameters must be taken care of simultaneously to infer sparse networks accurately. The newly proposed method achieves this by inducing both group and element sparse priors so that a sparse model structure is imposed, as well as the least number of candidate functions. EM algorithm is used to solve the Type II maximisation efficiently. This framework can also be applied to linear ARX models. In this case, element sparse priors penalise the polynomial order. Simulations also show our method is superior to SBL, GSBL and kernel methods.

Further developments should include two aspects. The first is to obtain theoretical guarantees of the algorithm performance. Since the dictionary matrix Φ correlates with process noise due to the intrinsic property of dynamic systems, its analysis is much more complex than a pure linear regression case [11]. The second question is how to extend this framework to infer more general network models, such as NARMAX models. The main obstacle here is that SBL normally demands the logarithm of the likelihood function be quadratic, which does not naturally happen with these model classes, i.e. their posterior distribution is intractable, even if the sparse prior is approximated by its lower bound.

REFERENCES

[1] L. Ljung, *System Identification: Theory for User*. Prentice Hall, 1998.

[2] A. Chiuso and G. Pillonetto, "A Bayesian approach to sparse dynamic network identification," *Automatica*, pp. 1553–1565, 2012.

[3] D. Wipf, "Bayesian methods for finding sparse representations," Ph.D. dissertation, University of California, 2006.

[4] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, vol. 22, pp. 231–245, 2013.

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

[6] D. Hayden, Y. Chang, J. Goncalves, and C. Tomlin, "Sparse network identifiability via compressed sensing," *Automatica*, vol. 68, pp. 9–17, 2016.

[7] B. Sanandaji, T. Vincent, and M. Wakin, "Exact topology identification of large-scale interconnected dynamical systems from compressive observations," *Proceedings of the 2011 American Control Conference*, pp. 649–656, 2011.

[8] S. Babacan, S. Nakajima, and M. Do, "Bayesian group-sparse modeling and variational inference," *Signal Process. IEEE Trans.*, vol. 62(11), pp. 2906–2921, 2014.

[9] M. Tipping, "Sparse Bayesian learning and the relevance vector mach," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[10] R. Chen, C. Chu, S. Yuan, and Y. Wu, "Bayesian sparse group selection," *Journal of Computational and Graphical Statistics*, pp. 1–29, 2015.

[11] D. Wipf, B. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Trans. Inf. Theory.*, vol. 57(9), pp. 6236–6255, 2011.

[12] W. Pan, Y. Yuan, J. Goncalves, and G. Stan, "A sparse Bayesian approach to the identification of nonlinear state-space systems," *IEEE Trans. Automat. Contr.*, vol. 61(1), pp. 182–187, 2016.

[13] W. Pan, Y. Yuan, L. Ljung, J. Goncalves, and G. Stan, "Identifying biochemical reaction networks using heterogeneous datasets," *IEEE Conference on Decision and Control*, 2015.

[14] G. Pillonetto and G. D. Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46(1), pp. 81–93, 2010.

[15] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50(3), pp. 657–682, 2014.

[16] T. Chen, M. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *Automatica*, vol. 59(11), pp. 1–33, 2014.

[17] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational em algorithms for non-Gaussian latent variable models," *Adv. Neural Inf. Process. Syst.*, vol. 18, p. 1059, 2006.

[18] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, pp. 1–305, 2008.

[19] Z. Zhang and B. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Trans. Signal Process.*, vol. 61(8), pp. 2009–2015, 2013.

[20] R. Giri and B. Rao, "Type I and type II Bayesian methods for sparse signal recovery using scale mixtures," *IEEE Transactions on Signal Processing*, pp. 3418–3428, 2016.

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

[22] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Transactions on Signal Processing*, vol. 55(7), pp. 3704–3716, 2007.