Bayesian topology identification of linear dynamic networks

Shengling Shi, Giulio Bottegal and Paul M. J. Van den Hof

***Abstract*— In networks of dynamic systems, one of the challenges is to identify the topology or interconnection structure of the network on the basis of measured node signals. Inspired by a Bayesian approach for estimating dynamic network models presented in [1], in this paper we explore a Bayesian model selection method for identifying the connectivity of networks of transfer functions, without the need to estimate the dynamics. The algorithm employs a Bayesian measure and a forward-backward search algorithm. To obtain the Bayesian measure, the impulse responses of network modules are modeled as Gaussian processes. The kernel hyperparameters are estimated by marginal likelihood maximization, using the expectation-maximization algorithm as a solver. Numerical results demonstrate the effectiveness of this method.**

I. INTRODUCTION

Estimation problems in system identification typically concern relatively simple structural setups, such as single-input-single-output (SISO) or multipe-input-multiple-output (MIMO) open-loop or closed-loop configurations [2]. Due to developments in data collection and computation, as well as in the increasing level of complexity of current days' technological systems, there is a strong need for the development of identification and estimation techniques in large-scale interconnected dynamic systems, usually referred to as dynamic networks.

Identification of networks of transfer functions involves multiple aspects, including the estimation of one local transfer function, which is called a module, in the network [3] [4] [5], estimation of the topology [6] [1], estimation of the full network model [1] [7] and identifiability aspects of the network models [8] [9] [10]. The network topology is sometimes assumed to be known in the estimation problems for dynamic networks [4] [11]. In many applications, estimation of the network topology is the main object of study, e.g. in systems biology [12], in social and political science [13], [14], or in finance [15]. In addition, the prediction accuracy of an internal variable in a dynamic network can sometimes be improved by using a model with sparse topology [16].

The problem of topology identification has been addressed using different approaches. Several methods making use of measures in the frequency domain can be found in [6] [17] [18]. The approach in [6] is based on the coherence function, which is built on the idea that nodes that are adjacent in a network should have a higher correlation than nodes that are more distant. However, this approach is developed for undirected tree structures only. A follow-up can be found in [17], where zero entries in a multivariate Wiener filter estimate of the dynamics are used to infer the topology. Using a non-causal Wiener filter, the network undirected graph may be recovered with extra edges that may not exist in the ground truth. The approach in [18] is formulated for continuous-time state-space models, building on the observation that the inverse of the cross spectrum matrix changes if a subset of signals are set to zero.

Some approaches make use of regularized regression to enforce a subset of parameters belonging to the same module to zero; the topology is then identified by the remaining nonzero parameters. Typical regularization strategies exploit the l_0 norm penalty [18] or the grouped version of the l_1 norm penalty [19] [20] [21] on the parameter vector.

Search algorithms have also been employed to estimate the topology. An iterative algorithm known as *block orthogonal matching pursuit* in compressed sensing employs a forward search procedure [22]. In the field of Bayesian networks, search algorithma coupled with Bayesian measures are commonly used to infer the topology [23]. However, the above approaches are typically not formulated for networks of transfer functions. A Bayesian approach formulated for dynamic networks can be found in [1], where the impulse responses of the modules are modeled as Gaussian processes whose kernel is parameterized by hyperparameters; these hyperparameters are modeled as random variables whose probability density aims at imposing a prior knowledge on the sparsity of the network.

Inspired by [1], in this paper a Bayesian model selection approach [24] [25] is explored to solve the topology identification problem. While in [1] focus was on the joint estimation of topology and dynamics, our aim is to develop a Bayesian approach for topology identification, without estimating the dynamics.

The Bayesian approach in this work employs a Bayesian measure coupled with a forward-backward search algorithm to select the topology which optimizes the measure. To obtain the Bayesian measure, a Gaussian prior distribution is assigned to the infinite impulse responses of the modules in dynamic networks. The hyperparameters of the prior are modeled as deterministic variables and estimated by maximizing marginal likelihood. The maximization of the likelihood is carried out using a computationally attractive instance of the expectation-maximization (EM) algorithm; this constitute a major difference from the approach in [1]. In addition, comparing to that work, in this paper the

This project has received funding from the European Research Council (ERC), Advanced Research Grant SYSDYNET, under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694504).

Shengling Shi, Giulio Bottegal and Paul M. J. Van den Hof are with the Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands {s.shi, g.bottegal, p.m.j.vandenhof}@tue.nl

topology is modeled as a random variable; this permits to incorporate structure prior information when required by specific applications.

This paper is organized as follows. In Section II, the problem of topology identification is formulated. In Section III, the Bayesian model selection is introduced and then extended to obtain a new algorithm for the problem of this work in Section IV. The group Lasso estimator and one variant of it are introduced in Section V to compare with the Bayesian approach. The numerical results are in Section VI and conclusions complete the paper. All proofs are collected in the Appendix.

II. Problem formulation

The linear dynamic network model first introduced in [3] is considered in this work:

$$w_j(t) = \sum_{i\in I\setminus j} G_{ji}(q)w_i(t) + H_j(q)e_j(t),\ j \in I, \quad (1)$$

where q^{-1} is the delay operator, i.e. $q^{-1}w_j(t) = w_j(t-1)$, $I = \{1, \cdots, L\}$ is the index set, G_{ji} is a transfer function and e_j is a white noise process. The notation w_I will be used to denote the set $\{w_j | j \in I\}$. With some abuse of notation, $w_j(t)$ denotes both a random variable and its realization. In addition, $Y \setminus B$ is used to denote the set difference between set Y and B, i.e. $Y \setminus B = \{x \in Y | x \notin B\}$.

Combining (1) into a matrix form, the full model can be written as

$$w(t) = G(q)w(t) + H(q)e(t),$$

where $w(t) = [w_1(t), ..., w_L(t)]^T$, $e(t) = [e_1(t), ..., e_L(t)]^T$ and $H(q)$ is a diagonal matrix containing $H_j(q)$. The matrix $G(q)$ contains $G_{ji}(q)$ and has zero entries on its main diagonal.

The assumptions on (1) are summarized here:

- $w_j(t)$ can be measured for all j and up to time N.
- $(I - G(q))^{-1}$ is proper and stable.
- $G_{ji}(q)$ is a stable and strictly proper rational transfer function, $H_j(q)$ is stable, monic and minimum-phase.
- $e_j(t)$ is a white noise process and is also independent over nodes j. $e_j(t)$ follows a Gaussian distribution with an unknown standard deviation σ_j:

$$e_j(t) \sim \mathcal{N}(0, \sigma_j^2),\ \forall t.$$

The topology of (1) can be defined as follows:

Definition 2.1: The topology $\mathcal{G}$ corresponding to (1) is defined as

$$\mathcal{G} = \{[i\ j] | G_{ji} \neq 0,\ i,\ j \in I\}.$$

The graphical representation of the topology is fully specified by $\mathcal{G}$, where the signals are represented by nodes and an directed edge $w_i \to w_j$ exists if $[ij] \in \mathcal{G}$. An example of a graphical representation can be found in Fig. 1, which shows the topology of the following network:

$$\begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & G_{13}(q) \\ G_{21}(q) & 0 & 0 \\ 0 & G_{32}(q) & 0 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix} + H(q)e(t). \quad (2)$$

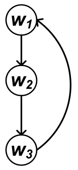


Fig. 1: Graph representation of a dynamic network.

The problem of topology identification is to identify $\mathcal{G}$ of the data generating system given the measurements of $w_j(t)$, for $t = 0, \cdots, N$ and all j. We shall denote such a set of measurement by D.

III. Bayesian model selection

In order to identify the topology, we need to define a measure that distinguishes two candidate structures on the basis of data. In this paper, a Bayesian model selection approach is employed, where the topology is modeled as a random variable [25]. To select one topology out of two candidates, the measure used to compare them is $P(\mathcal{G}_1|D)/P(\mathcal{G}_2|D)$, where $P(\mathcal{G}_i|D)$ is the posterior probability of $\mathcal{G}_i$ given data D. The measure can be further formulated as

$$\frac{P(\mathcal{G}_1|D)}{P(\mathcal{G}_2|D)} = \frac{P(D|\mathcal{G}_1)P(\mathcal{G}_1)}{P(D|\mathcal{G}_2)P(\mathcal{G}_2)} = \frac{P(D|\mathcal{G}_1)}{P(D|\mathcal{G}_2)}, \quad (3)$$

where $P(D|\mathcal{G})$ is the marginal likelihood and the second equality holds when the prior distribution of the topology is assumed to be uniform. $P(D|\mathcal{G}_1)/P(D|\mathcal{G}_2)$ is called *Bayes factor* [24]: taking the logarithm of $P(D|\mathcal{G})$, we can obtain an objective function whose maximization yields the topology with the highest marginal likelihood. Note that the Bayesian information criterion (BIC) is an approximation of $\log P(D|\mathcal{G})$ with a bounded error when $N \to \infty$ [24].

When the modules in the dynamic network are parameterized by a stochastic parameter vector θ, the marginal likelihood in (3) can be obtained as

$$P(D|\mathcal{G}) = \int P(D|\theta, \mathcal{G})P(\theta|\mathcal{G})d\theta, \quad (4)$$

where $P(D|\theta, \mathcal{G})$ is the likelihood and $P(\theta|\mathcal{G})$ is the parameter prior distribution. It can be seen that two important steps to apply (3) are the calculation of the integration and the choice of the parameter prior distribution in (4). In this work, it is assumed that there is no prior knowledge about the topology and thus $P(\mathcal{G}_i) = P(\mathcal{G}_j)$. For the reader who is interested in the structure prior, an example can be found in [26].

Following the Bayesian approach, the topology maximizing $\log P(D|\mathcal{G})$ is the solution of the problem under study. The associated optimization problem can be formulated as follows:

$$\max_{\mathcal{G}\in\mathcal{G}_{\text{set}}} \log P(D|\mathcal{G}), \quad (5)$$

where $\mathcal{G}_{\text{set}}$ denotes the set of all possible graphs. To deal with (5), we need to address i) the choice of $P(\theta|\mathcal{G})$ in (4), ii) the calculation of the integration in (4), and iii) the solver to select the topology when there are a large number of candidates.

IV. Bayesian topology identification

In this section, the issues that occur when solving (5) are discussed.

A. Reformulation of the problem

Model (1) can be reformulated as

$$w_j(t) = \hat{w}_j(t|t-1) + e_j(t), \tag{6}$$

where $\hat{w}_j(t|t-1)$ is the one-step ahead predictor, namely

$$\hat{w}_j(t|t-1) = [1 - H_j^{-1}(q)]w_j(t) + \sum_{i \in I \setminus j} \frac{G_{ji}(q)}{H_j(q)} w_i(t),$$

and

$$\frac{G_{ji}(q)}{H_j(q)} = \sum_{k=1}^{\infty} \theta_{ji,k} q^{-k},\ 1 - H_j^{-1}(q) = \sum_{k=1}^{\infty} \theta_{jj,k} q^{-k},$$

under the assumptions that $G_{ji}(z)$ is stable and $H_j(z)$ is minimum-phase [2]. The compact form of (6) containing measurements up to time N can be written as

$$w_j^N = \sum_{i \in I} A_{ji}\theta_{ji} + e_j^N, \quad j \in I \tag{7}$$

where $w_j^N = [w_j(1), ..., w_j(N)]^T$, $\theta_{ji} = [\theta_{ji,1}, ..., \theta_{ji,n}]^T$, $e_j^N = [e_j(1), ..., e_j(N)]^T$, and

$$A_{ji} = \begin{bmatrix} w_i(0) & w_i(-1) & \cdots & w_i(-n+1) \\ w_i(1) & w_i(0) & \cdots & w_i(-n+2) \\ \vdots & \ddots & \ddots & \vdots \\ w_i(N-1) & w_i(N-2) & \cdots & w_i(N-n) \end{bmatrix}.$$

Here, n denotes the model order. Note that the model order n is infinite in principle; however, an approximation to a sufficiently high integer will have no impact on the performance of the method. Equation (7) can also be written as $w_j^N = A_j\theta_j + e_j^N$, where $A_j = [A_{j1}, ..., A_{jL}]$ and $\theta_j = [\theta_{j1}^T, ..., \theta_{jL}^T]^T$. Equivalently, the problem considered in this work can be also formulated based on (7) as follows. Given the measurements of $w_j(t)$ for all j and up to time N, the problem of topology identification is to identify the set $\bar{\mathcal{G}} = \{[i\ j] | \theta_{ji} \neq 0,\ i,\ j \in I\}$. Note that $\bar{\mathcal{G}}$ is defined on the predictor model (6) while $\mathcal{G}$ is defined on (1). The self-loops, i.e. $w_j \to w_j$, may appear in $\bar{\mathcal{G}}$ to denote the dependence of $w_j(t)$ on its own past values in (6) while self-loops do not exist in $\mathcal{G}$ since G has zero entries on the main diagonal. Thus, $\mathcal{G}$ is equivalent to $\bar{\mathcal{G}}$ when the self-loops are removed and there is an one-to-one map between the two graphs. Even if the algorithm is designed to recover $\bar{\mathcal{G}}$, the notation $\mathcal{G}$ is still used in place of $\bar{\mathcal{G}}$, while the self-loops are made implicit to improve the readability of the topology.

B. Decomposition of the objective function

In this section, we show that the objective function $\log P(D|\mathcal{G})$ can be decomposed into a set of independent terms corresponding to MISO problems, where each MISO topology identification problem can be solved independently.

Based on (4), it can be seen that $P(D|\mathcal{G})$ can be factorized by decomposing $P(D|\theta, \mathcal{G})$ and $P(\theta|\mathcal{G})$. Due to the Bayes' rule and the assumption that the noises are white and independent over nodes, if each MISO model is independently parameterized, it holds that the likelihood can be factorized as

$$P(D|\theta, \mathcal{G}) = \prod_{j=1}^{L} \prod_{t=1}^{N} P(w_j(t)|\hat{w}_j(t|t-1)). \tag{8}$$

The independent parameter assumption implies that the term $P(\theta|\mathcal{G})$ in (4) satisfies

$$P(\theta|\mathcal{G}) = \prod_{j=1}^{L} P(\theta_j|\mathcal{G}_j), \tag{9}$$

where $\mathcal{G}_j$ and θ_j denote the topology and the parameter vector of one MISO model, respectively. Thus, given (1) and the global parameter independence assumption, the marginal likelihood in (5) can be decomposed into L independent terms as

$$\log P(D|\mathcal{G}) = \sum_{j=1}^{L} \log P(D_j|\mathcal{G}_j),$$

where D_j denotes the data relevant to a single MISO problem of the type (7) and

$$\log P(D_j|\mathcal{G}_j) \triangleq \log \int \prod_{t=1}^{N} P(w_j(t)|\hat{w}_j(t|t-1)) P(\theta_j|\mathcal{G}_j) d\theta_j. \tag{10}$$

Since each term is a function of the MISO topology, the search algorithm for the MISO topology can then be parallelized to obtain the overall network topology.

C. Objective function: Parameter prior and integration

Due to the independence among the MISO problems, in this section we describe the developed algorithm for a single MISO model of the type (7)

Firstly, we need to specify the dependence of $P(D_j|\theta_j, \mathcal{G}_j)$ and $P(\theta_j|\mathcal{G}_j)$ on one particular structure $\mathcal{G}_j$. Given one topology $\mathcal{G}_j = \{[i_1\ j], ..., [i_p\ j]\}$, $P(\theta_j|\mathcal{G}_j)$ considers the distribution of the parameter vector formulated based on $\mathcal{G}_j$, i.e. $\theta_j|\mathcal{G}_j = \begin{bmatrix} \theta_{ji_1}^T & \cdots & \theta_{ji_p}^T \end{bmatrix}^T$. Note that with some abuse of notation, $\theta_j|\mathcal{G}_j$ denotes a vector formulated based on the indexes in $\mathcal{G}_j$. In addition, the likelihood function $P(D_j|\theta_j, \mathcal{G}_j)$ is calculated based on the model $w_j^N = (A_j|\mathcal{G}_j) \times (\theta_j|\mathcal{G}_j) + e_j^N$, where $A_j|\mathcal{G}_j = \begin{bmatrix} A_{ji_1} & \cdots & A_{ji_p} \end{bmatrix}$.

Parameter prior: Following the kernel-based approach for system identification [27], since the prior knowledge that the impulse responses should decay with time is available, the parameter prior $P(\theta_j|\mathcal{G}_j)$ is chosen from [28] as

$$\theta_j|\mathcal{G}_j \sim \mathcal{N}(0, K_j), \tag{11}$$

where

$$K_j = \begin{bmatrix} \lambda_{ji_1}\bar{K}(\beta_{ji_1}) & 0 & \cdots & 0 \\ 0 & \lambda_{ji_2}\bar{K}(\beta_{ji_2}) & \ddots & \vdots \\ \vdots & \ddots & & 0 \\ 0 & \cdots & 0 & \lambda_{ji_p}\bar{K}(\beta_{ji_p}) \end{bmatrix},$$

$\bar{K}(\beta_{ji})$ is a $n \times n$ matrix and the (k,q) entry of $\bar{K}(\beta_{ji})$ is defined by $\beta_{ji}^{\max(k,q)}$. It is required that $\lambda_{ji} > 0$ and $\beta_{ji} \in [0\ 1)$. For this choice of kernel $\bar{K}$, β_{ji} regulates the velocity of the decay of the impulse responses. Therefore, the module priors depend on the unknown hyperparameter vectors, i.e. $\lambda_j|\mathcal{G}_j = \begin{bmatrix}\lambda_{ji_1} & \cdots & \lambda_{ji_p}\end{bmatrix}^T$ and $\beta_j|\mathcal{G} = \begin{bmatrix}\beta_{ji_1} & \cdots & \beta_{ji_p}\end{bmatrix}^T$. Since every MISO problem will be assigned an independent parameter prior as (11), equation (9) is satisfied.

Integration: Denote $\eta_j = \begin{bmatrix}\sigma_j & \lambda_j^T & \beta_j^T\end{bmatrix}^T$, where the dependencies of λ_j and β_j on $\mathcal{G}_j$ are implicit. Based on (7), (11) and (10), given one particular $\mathcal{G}_j$, (10) can be obtained in a closed form as

$$\begin{aligned}\log P(D_j|\mathcal{G}_j) = & -\frac{1}{2}(w_j^N)^T\Gamma_j^{-1}w_j^N - \frac{1}{2}\log\det\Gamma_j \\ & + \text{constant term},\end{aligned}$$

where $\Gamma_j = \sigma_j^2 I_N + A_j K_j A_j^T$ and the dependencies of A_j and K_j on a particular topology $\mathcal{G}_j$ are implicit. Note that Γ_j is also a function of η_j. Therefore, the objective function of optimization problem (5) can be written as

$$\begin{aligned}J(\mathcal{G}_j;\eta_j) &= 2\log P(D_j|\mathcal{G}_j;\eta_j) - \text{constant term} \\ &= -(w_j^N)^T\Gamma_j^{-1}w_j^N - \log\det\Gamma_j,\end{aligned} \tag{12}$$

where we have highlighted its dependence upon the hyperparameter vector η_j. Since η_j is unknown, an estimate of η_j has to be computed first. Then, we can use $J(\mathcal{G}_j;\hat{\eta}_j)$ as the objective function for the topology estimation problem.

Estimation of hyperparameters: To obtain an estimate of $\hat{\eta}_j$, we estimate the hyperparameter vector associated to the full graph, namely η_j^{full}. Then, given a graph $\mathcal{G}_j$, the corresponding hyperparameter vector $\hat{\eta}_j$ associated to that graph can be obtained by neglecting those hyperparameters associated to zero modules (i.e., missing edges in the graph). This procedure avoids the re-estimation of η_j for all different graphs and reduces the computational cost. The hyperparameter vector η_j^{full} is estimated by solving the following marginal likelihood problem:

$$\hat{\eta}_j^{full} = \underset{\eta_j^{full}}{\operatorname{argmax}} \log P(D_j|\mathcal{G}_j^{full};\eta_j^{full}), \tag{13}$$

where $\mathcal{G}_j^{full}$ is a full graph, i.e. $\mathcal{G}_{j,full} = \{[1\ j],...,[L\ j]\}$. A local optimum of this problem can be found by the EM algorithm [29].

Assuming that an estimate $\hat{\eta}_j^{(k)}$ of η_j^{full} is available at the k-th iteration of the EM algorithm, an update estimate is obtained by the following steps:

(**E-step**) Compute

$$Q(\eta_j, \hat{\eta}_j^{(k)}) = E_{P(\theta_j|w_j^N;\hat{\eta}_j^{(k)})}[\log P(\theta_j, w_j^N;\eta_j)]; \tag{14}$$

(**M-step**) Compute

$$\hat{\eta}_j^{(k+1)} = \underset{\eta_j\in\mathcal{V}}{\operatorname{argmax}}\, Q(\eta_j, \hat{\eta}_j^{(k)}), \tag{15}$$

Note that for a MISO problem, the input and the graph are regarded as fixed and thus implicit in (14).

Proposition 4.1: Denote $\hat{\eta}^{(k)}$ as the estimate of the hyperparameter vector at the kth iteration of the EM algorithm used to solve (13). Then, according to (14) and (15), $\hat{\eta}^{(k+1)}$ is obtained with the following update rules:

- The hyperparameter $\hat{\sigma}_j^{k+1}$ is obtained as

$$\hat{\sigma}_j^{k+1} = \sqrt{\frac{M^{(k)}}{N}}, \tag{16}$$

where

$$\begin{aligned}M^{(k)} =&(w_j^N)^T w_j^N - 2(w_j^N)^T A_j \hat{C}_j^{(k)} w_j^N \\ &+ tr[A_j^T A_j \hat{\Delta}_j^{(k)}], \\ \hat{C}_j^{(k)} =&[\hat{\sigma}_j^{(k)}]^{-2}[\hat{\Sigma}_j^{(k)}]^{-1}A_j^T, \\ \hat{\Sigma}_j^{(k)} =&[\hat{\sigma}_j^{(k)}]^{-2}A_j^T A_j + [K_j(\hat{\lambda}_j^{(k)}, \hat{\beta}_j^{(k)})]^{-1}, \\ \hat{\Delta}_j^{(k)} =&[\hat{\Sigma}_j^{(k)}]^{-1} + \hat{C}_j^{(k)} w_j^N (w_j^N)^T [\hat{C}_j^{(k)}]^T.\end{aligned}$$

- The hyperparameter $\hat{\beta}_{ji}^{k+1}$, $i = 1,...,L$, is obtained as

$$\begin{aligned}\hat{\beta}_{ji}^{k+1} = \arg\min_{\beta_{ji}\in[0\ 1)} & n\log[tr(\bar{K}^{-1}(\beta_{ji})\hat{\Delta}_j^{(k)}[i])] \\ & + \log\det\bar{K}(\beta_{ji}),\end{aligned} \tag{17}$$

where $\hat{\Delta}_j^{(k)}[i])$ is a square sub-matrix obtained from $\hat{\Delta}_j^{(k)}$ by the $[(i-1)n+1]$-th row and column until the (in)-th row and column of $\hat{\Delta}_j^{(k)}$.

- The hyperparameter $\hat{\lambda}_{ji}^{k+1}$, $i = 1,...,L$, is obtained as

$$\hat{\lambda}_{ji}^{k+1} = \frac{1}{n}tr[\bar{K}^{-1}(\hat{\beta}_{ji}^{k+1})\hat{\Delta}_j^{(k)}[i]]. \tag{18}$$

It can be found that (13) is decomposed into a set of optimization problems with scalar optimization variables for estimating β and closed-form solutions for estimating σ and λ. The computational speed of the above algorithm can be further improved by exploiting the factorization of $\bar{K}$ [30] [29], which is also implemented in the algorithm.

D. Algorithm for optimization

The objective function of problem (5) has been formulated as shown in (12), where $J(\mathcal{G}_j;\hat{\eta}_j)$ is instead used without changing the result of the optimization and $\hat{\eta}_j$ is obtained as $\hat{\eta}_j = \hat{\eta}_j^{full}|\mathcal{G}_j$. The next step is to design the solver for the optimization problem.

Since the number of all possible directed graphs in $\mathcal{G}_{\text{set}}$ is 2^{L^2-L}, it is infeasible to consider all the candidates. Following [23], a forward-backward greedy search algorithm is implemented to find a local optimum of (5). Recall that the graph of the predictor model is considered here, so that self-loops are generally present. The algorithm initializes a graph with only self-loops and then starts the edge-addition phase, where at each iteration, the edge which most improves the objective value is added to the graph from the previous iteration. The iterations stop when no improvement can be found by adding edges.

Given the final graph of the edge-addition phase, the algorithm starts the edge-deletion phase, where at each iteration, one edge is removed from the graph of the previous iteration if such deletion improves the objective function comparing to the removal of other edges. The final output of the algorithm is obtained when no improvement in the objective value can be found by deleting any edge.

As mentioned earlier, due to the decomposition in (10), the search algorithm can be applied to every MISO problem separately, merging the outcomes to obtain the network topology.

E. Final algorithm

After the formulation of the objective function and the greedy search algorithm, the algorithm is now complete and summarized in this section. Firstly, recall that $\hat{\eta}_j^{full}$ obtained in the previous step is for a full graph and thus, given a structure $\mathcal{G}_j$, $\hat{\eta}_j$ should be reformulated as $\hat{\eta}_j^{full}|\mathcal{G}_j = \begin{bmatrix}\hat{\sigma}_j & (\hat{\lambda}_j^{full}|\mathcal{G}_j)^T & (\hat{\beta}_j^{full}|\mathcal{G}_j)^T\end{bmatrix}^T$. To simplify the notation, the index j is dropped in the algorithm.

Algorithm (BS - Bayesian Search): Inputs: data D; Outputs: $\hat{\mathcal{G}}$

```
1) Obtain $\hat{\eta} = \max_\eta \log p(D|\mathcal{G}_{full};\eta)$ by EM algorithm
2) Initialize $\mathcal{G}^{(0)} = \{[j\,j]\}$ and $Edge = \{[1\,j], \cdots, [L\,j]\}$
3) For $b = 1 : L-1$ (Edge-addition phase)
   • $Edge = Edge \setminus \hat{\mathcal{G}}^{(b-1)}$
   • $[\hat{i}\,j] = arg\max_{[i\,j]\in Edge} J(\{\hat{\mathcal{G}}^{(b-1)},\ [i\,j]\};\hat{\eta})$
   • if $J(\{\hat{\mathcal{G}}^{(b-1)},\ [\hat{i}\,j]\};\hat{\eta}) - J(\hat{\mathcal{G}}^{(b-1)};\hat{\eta}) > \tau$
       ◇ $\hat{\mathcal{G}}^{(b)} = \{\hat{\mathcal{G}}^{(b-1)},\ [\hat{i}\,j]\}$
   • else
       ◇ break loop
4) Initialize for the second phase: $\hat{\mathcal{G}}^{(0)} = \hat{\mathcal{G}}_{FinalAddition}$
5) For $d = 1 : |\hat{\mathcal{G}}^{(0)}|$ (Edge-deletion phase)
   • $[\hat{i}\,j] = arg\max_{[i\,j]\in\hat{\mathcal{G}}^{(d-1)}} J(\hat{\mathcal{G}}^{(d-1)} \setminus [i\,j];\hat{\eta})$
   • if $J(\hat{\mathcal{G}}^{(d-1)} \setminus [\hat{i}\,j];\hat{\eta}) - J(\hat{\mathcal{G}}^{(d-1)};\hat{\eta}) > \tau$
       ◇ $\hat{\mathcal{G}}^{(d)} = \hat{\mathcal{G}}^{(d-1)} \setminus [\hat{i}\,j]$
   • else
       ◇ break loop
```

The tolerance τ, determining whether an edge should be added or removed (depending on the phase of the algorithm), is chosen to be $\tau = 0$ as default value; its suggested range is $[0, 10]$, see [24].

Remark 1: To empirically validate the choice of using the estimate of $\hat{\eta}^{full}$ under the full graph, the BS algorithm is compared with its variant using an iterative EM approach, which re-estimates $\hat{\eta}$ by the EM algorithm under every iteration of the search algorithm. We call this procedure the iterative-EM BS algorithm. ■

Comparing to the approach in [1], the main difference of the BS algorithm is that the hyperparameters are modeled as deterministic variables and then estimated by the EM algorithm. By contrast, in [1], the hyperparameters are modeled as random variables and a prior distribution of the hyperparameters is also used. The choice of modeling also the hyperparameters as random variables requires designing their prior distribution, which usually requires to include additional hyper-hyperparameters that may be difficult estimate.

V. Kernel-based group Lasso

The performance of the BS algorithm is compared with the group Lasso (GLasso) estimator [19], which is formulated on the basis of (7) as

$$\min_{\theta_j} \frac{1}{2}\left\| w_j^N - A_j\theta_j \right\|_2^2 + \delta_j \sum_{i=1}^{L} \left\|\theta_{ji}\right\|_2 . \qquad (19)$$

Here, the topology estimation problem is also divided in MISO problems to be solved independently, in order to reconstruct the final graph. It is also of interest to see if the performance of (19) can be improved by incorporating of the covariance matrix in (11) into the regularization term. This kernel-based GLasso, first introduced in [1], can be formulated as

$$\min_{\theta_j} \frac{1}{2}\left\| w_j^N - A_j\theta_j \right\|_2^2 + \delta_j \sum_{i=1}^{L} \sqrt{\theta_{ji}^T \bar{K}(\beta_j, n)^{-1}\theta_{ji}}. \qquad (20)$$

For simplicity, we choose to have the same hyperparameters β_j for all the modules of each MISO problem, in order to reduce the computational complexity. In order to select δ_j and β_j, cross validation can be employed. After having the estimated parameters, the topology can be obtained by checking if the l_2 norm of the parameter vector corresponding to one module is zero.

VI. Numerical results

To evaluate the performance of the algorithms, an existing edge in the network is labeled as one positive instance; its absence is labeled as one negative instance. Let P denote the total number of positives and N denote the total number of negatives in the ground truth. In addition, for the outcome of the algorithm, if the algorithm outputs one edge that does exist in the ground truth, it scores a true positive (TP). If the algorithm outputs one edge That does not exist in the ground truth, it scores a false positive (FP). The behavior of the algorithms is studied by using the receiver operating characteristic (ROC) curve [31], i.e. TP rate (TPR) vs FP rate (FPR) over different values of their tuning parameters, where

$$TPR = \frac{TP}{P}, \quad FPR = \frac{FP}{N},$$

which are further averaged over the number of Monte Carlo experiments. The tuning parameter for the BS algorithm and the iterative-EM BS algorithm is $\tau \in \{0, 1, ..., 10\}$, while the tuning parameters of GLasso and the kernel-based GLasso are $\delta_j \in \{0, 10, 20, ..., 2000\}$ and $\beta_j = \{0.1, ..., 0.9\}$. To build ROC curves for the two GLasso estimators, δ_j and β_j are kept the same for all MISO problems to reduce the number of tuning parameters. The $(0, 1)$ point in the ROC plot denotes the ideal performance without any error. Thus, the points on ROC curves of different methods can be compared based on their closeness to the $(0, 1)$ point, i.e. computing $dis = \sqrt{FPR^2 + (1 - TPR)^2}$. A smaller dis value implies a better performance.

Each experiment is conducted on the basis of a 6-node network with random topology and random, independent transfer functions. Three studies are conducted to construct the ROC curves, where every study contains 50 random experiments. In addition, $N = 2000$ and $n = 100$ for the first study, $N = 500$ and $n = 100$ for the second study, while $N = 50$ and $n = 50$ for the last study. Note that in the final study, the number of the postulated unknown parameters in the algorithm is larger than the number of the measurements. More details about the data generation can be found in the Appendix.

The obtained ROC curves are summarized in Fig 2. It can be found that in all tests, the two search algorithms perform better than the two GLasso estimators because the ROC curves of the search algorithms are closer to the $(0, 1)$ point for every value of τ. To compare the performance of the iterative-EM BS and the BS algorithm, the following measure is used:

$$V = [\sum_{i=1}^{11} \frac{dis_{iter-EMBS,i} - dis_{BS,i}}{dis_{BS,i}}] \div 11 \times 100\%,$$

where i denotes the ith value of τ in $\{0, 1, ..., 10\}$. Given one value of τ, one point on the ROC curve is correspondingly selected and thus $dis_{BS,i}$ can be calculated based on Fig 2. Note that a positive value of V implies a worse performance of the iterative-EM algorithm. It can then be found that $V = -33\%$ when $N = 2000$, $V = 2.5\%$ when $N = 500$ and $V = 27\%$ when $N = 50$. Thus, in the tests, the iterative-EM BS algorithm performs better than the BS algorithm when the sample size is large while it has worse performance when the sample size is relatively small. Intuitively, this can be explained by the fact that the iterative-EM algorithm relies more on the data because it adjusts the parameter prior given every different graph during the search procedure, leading to a larger error when the data length is limited. The computational speed of the iterative-EM algorithm is also around 10 times slower in this 6-node example. Thus, it is suggested to use the BS algorithm when N is small and the faster computation is preferred.

The performance of the algorithms is also compared when cross validation is employed for the two GLasso estimators while τ equals to the default value, i.e. $\tau = 0$, for the two BS algorithms. For the cross validation of the two GLasso estimators, the training data contains the data up to time $2(N+1)/3$ and the data left is kept for validation. The tuning parameter that provides the smallest root-mean-square error in predicting the validation data is selected. Note that in this case, the tuning parameters of the two GLasso estimators are allowed to be different over the MISO problems. The final results contain one (FPR, TPR) point for every algorithm and their distance to $(0, 1)$ is summarized in Table I.

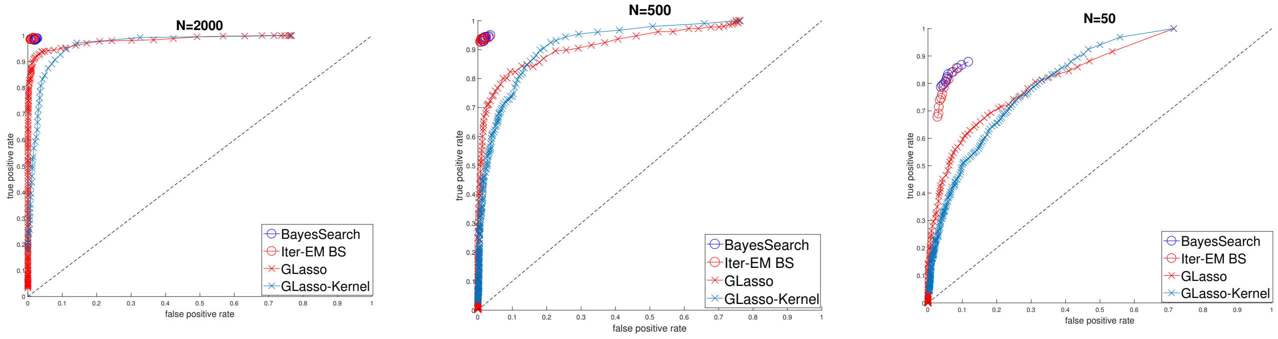
TABLE I: Distance of the results of the algorithms to $(0, 1)$ with the cross-validated or the default tuning parameter

	BS	Iter-EM BS	GLasso	K-GLasso
$N = 2000$	0.03	0.02	0.64	0.66
$N = 500$	0.06	0.06	0.39	0.59
$N = 50$	0.17	0.17	0.43	0.51

It can be found that the two search algorithms outperform the two GLasso estimators due to their smaller distance to $(0, 1)$. This is because the cross validation is designed for obtaining the tuning parameters corresponding to the best prediction performance, which typically leads to a model with more positives to improve the prediction. Instead, the Bayes factor typically favors simpler models, which may lead to a model with poorer prediction performance. This difference in the design purpose between BIC, which is an asymptotic approximation of the Bayes factor, and cross validation is also mentioned in [32].

VII. Conclusion

A Bayesian approach for topology identification of networks of transfer functions is explored. It uses the Bayes factor coupled with a forward-backward search algorithm. The Bayes factor is obtained by modeling the infinite impulse responses of the modules as Gaussian processes, where the hyperparameters of the Gaussian prior are estimated by the EM algorithm. Numerical results demonstrate the effectiveness of the algorithm, which shows better performance compared to the group Lasso estimator.



(a)

(b)

(c)

Fig. 2: TPR vs FPR over tuning parameters for data sets in different sizes: $N = 2000$ (left), $N = 500$ (middle), $N = 50$ (right)

APPENDIX

A. Proof of Proposition 4.1

The proof contains two steps, including the E-step and the M-step of the EM algorithm.

Proof: **E-step**: Firstly, note that $\log P(\theta_j, w_j^N; \eta_j) = \log P(w_j^N|\theta_j; \eta_j) + \log P(\theta_j; \eta_j)$, where $\log P(w_j^N|\theta_j; \eta_j)$ is the likelihood function given by the model and $\log P(\theta_j; \eta_j)$ is the parameter prior of the full graph given by (11). Thus, it can be found that

$$\begin{aligned}\log P(\theta_j, w_j^N; \eta_j) =&\text{constant} - \frac{1}{2}\sum_{i=1}^{L} \log\det(\lambda_{ji}\bar{K}(\beta_{ji}))\\ &- \frac{1}{2}\log\det(\sigma_j^2 I_N) - \frac{1}{2\sigma_j^2}(w_j^N)^T w_j^N\\ &- \frac{1}{2}\theta_j^T \Sigma_j \theta_j + \frac{1}{\sigma_j^2}(w_j^N)^T A_j \theta_j.\end{aligned}$$

$Q(\eta_j, \hat{\eta}_j^{(k)})$ can then be obtained by calculating the expectation of $\log P(\theta_j, w_j^N; \eta_j)$ over the posterior distribution of θ_j given the data and $\hat{\eta}^{(k)}$. Due to the Gaussian noise and the parameter prior (11), it follows that the posterior distribution of the parameter also has a Gaussian distribution as

$$\theta_j | w_j \sim \mathcal{N}(\hat{C}_j^{(k)} w_j^N, (\hat{\Sigma}_j^{(k)})^{-1}).$$

Thus, the E-step can be finalized as

$$Q(\eta_j, \hat{\eta}_j^{(k)}) = Q_1(\sigma_j, \hat{\eta}_j^{(k)}) + \sum_{i=1}^{L} Q_2(\lambda_{ji}, \beta_{ji}, \hat{\eta}_j^{(k)}) + \text{constant}, \tag{21}$$

where

$$\begin{aligned}Q_1(\sigma_j, \hat{\eta}_j^{(k)}) =& -\frac{1}{2}tr(\sigma_j^{-2} A_j A_j \hat{\Delta}_j^{(k)}) - N\log\sigma_j\\ &+ \frac{1}{\sigma_j^2}(w_j^N)^T A_j \hat{C}_j^{(k)} w_j\\ &- \frac{1}{2\sigma_j^2}(w_j^N)^T w_j^N,\\ =& -N\log\sigma_j - \frac{1}{2\sigma_j^2}M^{(k)}, \end{aligned}\tag{22}$$

$$\begin{aligned}Q_2(\lambda_{ji}, \beta_{ji}, \hat{\eta}_j^{(k)}) =& -\frac{1}{2}\log\det[\lambda_{ji}\bar{K}(\beta_{ji})]\\ &- \frac{1}{2}tr[(\lambda_{ji}\bar{K}(\beta_{ji}))^{-1}\hat{\Delta}_j^{(k)}[i]].\end{aligned}\tag{23}$$

where $M^{(k)}$ is formulated as shown in (16).

It can be found that Q is decomposed into two parts, including Q_1 as a function of σ_j and Q_2 as a function of the parameters from the parameter prior. Thus, the optimization of Q can be solved by considering Q_1 and Q_2 independently. The constant term in (21) will be ignored because it does not influence the optimization result.

M-step: It can be found that (22) is maximized by (16) assuming that $M^{(k)} > 0$.

To maximize $Q_2(\lambda_{ji}, \beta_{ji}, \hat{\eta}^{(k)})$, set the derivative of (23) over λ_{ji} to be zero, which leads to the solution of λ_{ji} as

$$\lambda_{ji}^* = \frac{1}{n}tr[\bar{K}^{-1}(\beta_{ji})\hat{\Delta}_j^{(k)}[i]], \tag{24}$$

which is a function of β_{ji}. Plugging (24) back into (23), one obtains that

$$\begin{aligned}Q_2(\lambda_{ji}^*, \beta_{ji}, \eta^{(k)}) =& -\frac{n}{2}\log[tr(\bar{K}^{-1}(\beta_{ji})\hat{\Delta}_j^{(k)}[i])]\\ &- \frac{1}{2}\log\det\bar{K}(\beta_{ji}) + \text{constant},\end{aligned}$$

which can be maximized by minimizing (17). After obtaining $\hat{\beta}_{ji}^{(k+1)}$, $\hat{\lambda}_{ji}^{(k+1)}$ can be found by (18). Thus, Q_1, Q_2 have been optimized independently and M-step is proved. ■

B. Data generation for the numerical results

For each experiment, the topology of the 6-node dynamic network, which is represented by a matrix with zeros on the main diagonal, is randomly generated with ones to represent existing edges and zeros for the non-existing edges.

For every edge, a discrete-time transfer function G_{ji} is randomly generated such that it is strictly proper, stable and has a random order ranging from 2 to 5. To guarantee a reasonable signal-to-noise ratio, G_{ji} is normalized by its own l_2 norm and then used to generate the data.

In addition, the noise model H_j is also randomly generated such that it is stable, monic and minimum-phase with a random order ranging from 2 to 5. The noise is sampled from a Gaussian distribution with zero mean and $\sigma = 1$.

REFERENCES

[1] A. Chiuso and G. Pillonetto, "A bayesian approach to sparse dynamic network identification," *Automatica*, vol. 48, no. 8, pp. 1553–1565, 2012.

[2] L. Ljung, *System Identification: Theory for the User*. Prentice-hall, 1987.

[3] P. M. J. Van den Hof, A. Dankers, P. S. C. Heuberger, and X. Bombois, "Identification of dynamic models in complex networks with prediction error methodsbasic methods for consistent module estimates," *Automatica*, vol. 49, no. 10, pp. 2994–3006, 2013.

[4] A. Dankers, P. M. J. Van den Hof, X. Bombois, and P. S. C. Heuberger, "Identification of dynamic models in complex networks with prediction error methods: Predictor input selection," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 937–952, 2016.

[5] J. Linder and M. Enqvist, "Identification and prediction in dynamic networks with unobservable nodes," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 574–10 579, 2017.

[6] D. Materassi and G. Innocenti, "Topological identification in networks of dynamical systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1860–1871, 2010.

[7] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Prediction error identification of linear dynamic networks with rank-reduced noise," *Automatica*, vol. 98, pp. 256–268, December 2018. [Online]. Available: https://arxiv.org/abs/1711.06369

[8] ——, "Identifiability of linear dynamic networks," *Automatica*, vol. 89, pp. 247–258, 2018.

[9] M. Gevers, A. S. Bazanella, and A. Parraga, "On the identifiability of dynamical networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 580–10 585, 2017.

[10] J. M. Hendrickx, M. Gevers, and A. S. Bazanella, "Identifiability of dynamical networks with partial node measurements," *ArXiv Preprint arXiv:1803.05885*, 2018.

[11] M. Gevers and A. S. Bazanella, "Identification in dynamic networks: Identifiability and experiment design issues." in *CDC*, 2015, pp. 4005–4010.

[12] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke, "Gene regulatory network inference: data integration in dynamic modelsa review," *Biosystems*, vol. 96, no. 1, pp. 86–103, 2009.

[13] W. M. Lord, J. Sun, N. T. Ouellette, and E. M. Bollt, "Inference of causal information flow in collective animal behavior," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 107–116, 2016.

[14] D. D. Zhang, H. F. Lee, C. Wang, and B. e. a. Li, “The causality analysis of climate change and large-scale human crisis,” *Proceedings of the National Academy of Sciences*, p. 201104268, 2011.

[15] M. Kar, Ş. Nazlıoğlu, and H. Ağır, “Financial development and economic growth nexus in the mena countries: Bootstrap panel granger causality analysis,” *Economic Modelling*, vol. 28, no. 1-2, pp. 685–693, 2011.

[16] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer series in statistics New York, NY, USA:, 2001, vol. 1, no. 10.

[17] D. Materassi and M. V. Salapaka, “On the problem of reconstructing an unknown topology via locality properties of the wiener filter,” *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1765–1777, 2012.

[18] S. Shahrampour and V. M. Preciado, “Topology identification of directed dynamical networks via power spectral analysis,” *IEEE Transactions on Automatic Control*, vol. 60, no. 8, pp. 2260–2265, 2015.

[19] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[20] A. Bolstad, B. D. Van Veen, and R. Nowak, “Causal network inference via group sparse regularization,” *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, 2011.

[21] Y. Wang, M. Sznaier, and O. Camps, “A super-atomic norm minimization approach to identifying sparse dynamical graphical models,” in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 1962–1967.

[22] P. Kuppinger, Y. C. Eldar, and H. Bölcskei, “Block-sparse signals: Uncertainty relations and efficient recovery,” *IEEE Transanctions on Signal Processing*, vol. 58, no. 6, 2010.

[23] D. M. Chickering, “Optimal structure identification with greedy search,” *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 507–554, 2002.

[24] R. E. Kass and A. E. Raftery, “Bayes factors,” *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773–795, 1995.

[25] L. Wasserman, “Bayesian model selection and model averaging,” *Journal of Mathematical Psychology*, vol. 44, no. 1, pp. 92–107, 2000.

[26] A. V. Werhli and D. Husmeier, “Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge,” *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, 2007.

[27] G. Pillonetto and G. De Nicolao, “A new kernel-based approach for linear system identification,” *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.

[28] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, “Kernel methods in system identification, machine learning and function estimation: A survey,” *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[29] G. Bottegal, A. Y. Aravkin, H. Hjalmarsson, and G. Pillonetto, “Robust em kernel-based methods for linear system identification,” *Automatica*, vol. 67, pp. 114–126, 2016.

[30] F. P. Carli, “On the maximum entropy property of the first-order stable spline kernel and its implications,” in *Control Applications (CCA), 2014 IEEE Conference on*. IEEE, 2014, pp. 409–414.

[31] D. Marbach, J. C. Costello, R. Küffner, *et al.*, “Wisdom of crowds for robust gene network inference,” *Nature Methods*, vol. 9, no. 8, p. 796, 2012.

[32] A. Gelman, J. Hwang, and A. Vehtari, “Understanding predictive information criteria for bayesian models,” *Statistics and Computing*, vol. 24, no. 6, pp. 997–1016, 2014.