Internship Report, Eindhoven University of Technology, 31 December 2019.

An iterative algorithm for learning dynamic networks with correlated noise

V. C. Rajagopal*, K. R. Ramaswamy†, P. M. J. Van den Hof†
*Student, Department of Electrical Engineering, Eindhoven University of Technology
†Supervisor, Department of Electrical Engineering, Eindhoven University of Technology

***Abstract*—Identification of dynamic networks poses a wide range of challenges and many studies have been conducted for handling these challenges. The current study considers one such problem, namely identification of a local module in a dynamic network with correlated noise. Recent studies in this domain formulate the identification as a Multiple Input Multiple Output (MIMO) estimation problem, parameterizing all the modules in the setup. This study however aims to reduce the number of parameters of estimation by representing all modules except the desired module in the estimation problem as zero mean independent gaussian processes with a covariance matrix given by first order stable spline kernel. The parameters of the module are obtained by maximizing the marginal likelihood of the output using the Empirical Bayes (EB) approach. The marginal likelihood maximization is approached through an Expectation Maximization (EM) algorithm.**

***Index Terms*—Empirical Bayes, Dynamic networks, Correlated noise, Expectation Maximization, Gaussian Process**

I. Introduction

Systems are becoming increasingly complex due to interconnections between subsystems. Such interconnected systems exist everywhere around us like power generation grids and brain networks. These systems with interconnections are termed as dynamic networks [8]. A dynamic network is considered as a set of measurable signals interconnected through linear dynamic systems, possibly driven by external excitation. These dynamic networks pose two key challenges, one is identification of the dynamics in the network, while the other is control of the dynamics. Considerable attention has been devoted to both these challenges, particularly in the area of identification of dynamics from the measured data. Previous literature corresponding to identification can be classified into three sub-problems. The first sub-problem is identification of the interconnection structure of the network (topology identification) which is discussed in [14], [13] and [5]. The second sub-problem is identification of the full network dynamics which is discussed in [19] and [10]. The last sub-problem is identification of a desired set of dynamic sub-systems (or modules) in a network. This problem is termed as local module identification.

Under the assumption of known network topology, the problem of local module identification has been dealt with in [8]. In this work, another assumption has been made that all process noise present in the network are uncorrelated to each other. With a configuration of networks based on these assumptions, [8] provides a method for identification of the desired module based on prediction error method. This method is an extension of the direct method for closed loop to the dynamic network identification and is, therefore, named as the direct method for general network topology. The direct method considers one output influenced by multiple inputs, consisting of the desired and additional modules, making it a Multiple Input Single Output (MISO) framework. Under this framework, additional modules are identified simultaneously along with the desired module. These additional modules result in a high number of parameters to be estimated. Prior to the estimation, a model order selection step using complexity criteria like AIC, BIC or Cross-Validation [11] should be carried out to select the number of parameters.

To avoid such complications, an algorithm that uses regularized kernel-based methods for estimation of modules in the direct method is presented in [17]. This work uses independent Gaussian processes to model the impulse responses of the additional modules and effectively reduce the number of parameters that need to be estimated. The desired module is represented using a parametric model. The method proposed in [17] is an Empirical Bayes (EB) method [12] motivated by the EB methods presented in [2], and [9]. The method in [2] can be used for identifying Single Input Single Output (SISO) models, while the method in [9] can be used for identifying dynamic networks without any process noise.

For dynamic networks where process noise is present on different nodes and are possibly correlated, [18] presents multiple algorithms on selecting predictor input and predicted output signals for estimation of the local module using a prediction error identification method. The prediction error method deals with confounding variables [6], where the correlated process noise directly and/or indirectly influence both the predicted inputs and predictor outputs. As a result, the algorithms presented in [18] mitigate the effect of confounding variables by extending the number of predictor inputs and/or predicted outputs to the identification problem. This in turn results in the estimation becoming a Multiple Input Multiple Output (MIMO) identification problem. The algorithms and method developed in [18] are termed as the local direct method. The local direct method faces the same complications that were present in the direct method but on a larger scale. The additional modules that need to be estimated are larger in number because of larger number of outputs. Carrying out a model order selection step for the MIMO estimation problem is complicated due to a large number of modules. In the example in figure 1, a model order selection step for the four modules with FIR model structure of orders

1 to 5 result in 4^5 combinations to test. For larger networks, this selection step becomes computationally infeasible. In addition to model order selection, parameterizing all the modules in the local direct method result in a large number of parameters. As the number of parameters in the estimation problem increase, the minimum variance that can be achieved for the parameters reduce. This reduction in variance is due to parameters of the additional module which are of minimal interest to the local module identification. As a result, this method requires an algorithm that simplifies the MIMO estimation problem. Although, [16] presents a work on non-parametric identification of all the modules in a MIMO system using regularized kernel methods, the local direct method requires an algorithm that estimates both parametric and non-parametric models.

Therefore, we formulate the following research question: *How to develop an algorithm that estimates the target module in a dynamic network with correlated process noise and overcomes the following challenges.*

- *Model order selection of the modules in the MIMO estimation problem.*
- *Reduce the number of parameters to be estimated (to reduce variance of target estimate).*

In this report, development of such an algorithm is discussed. Using the algorithms of [18], the predictor inputs and prediction outputs are selected and the MIMO structure is written for identification using the prediction error method. Then, the desired module is represented as a parametric model in order to obtain an accurate description of its dynamics. In case of the additional modules of the MIMO structure, their impulse responses are modelled as zero mean independent Gaussian processes. The covariance matrices of these Gaussian vectors are modelled using first-order stable spline kernels, ensuring stability and smoothness to the impulse responses. The effect of process noise is also incorporated indirectly in the impulse response modelling. Using this approach, we obtain a Gaussian probabilistic description depending on a vector of parameters η, containing the parameters of the desired module, the covariance of process noise and the hyperparameters governing the stable spline kernel. To obtain the parameters of the desired module, η is estimated. To accomplish this task, we consider an EB approach, where η is obtained by maximizing the marginal likelihood of the data. The solution to this optimization is achieved through an iterative solution scheme based on Expectation Maximization (EM) [7] algorithm.

II. Problem statement

We consider the problem setting defined in [8]. Following this setting, a dynamic network that is built up of L nodes is considered with $w_j(t)$, $j = 1, \dots, L$. The network is defined by the equation,

$$w_j(t) = \sum_{\substack{l=1 \\ l \neq j}}^{L} G_{jl}(q) w_l(t) + u_j(t) + v_j(t) \tag{1}$$

where, $j \in \{1, \dots, L\}$ and q^{-1} is the delay operator.

- G_{jl} are strictly proper rational transfer functions referred to as *modules*,
- There are no self loops in the system i.e. nodes are not directly related to itself, $G_{jj} = 0$,
- v_j is process noise acting on the node j. It is noted that the $v = \begin{bmatrix} v_1 & \dots & v_L \end{bmatrix}^T$ is modelled as a stationary stochastic process with rational spectral density, $\Phi_v(\omega)$, such that there exists a $e = \begin{bmatrix} e_1 & \dots & e_L \end{bmatrix}^T$ with covariance, $\Lambda > 0$ such that, $v(t) = H(q)e(t)$, where $H(q)$ is a stable, monic and minimum phase transfer function matrix.
- $u_j(t)$ is generated by *external variables* $r_k(t)$, that can be directly manipulated by the user and is given by $u_j(t) = \sum_{k=1}^{K} R_{jk} r_k(t)$, where R_{jk} are stable and proper transfer functions. Therefore, $u(t) = \begin{bmatrix} u_1(t) & \dots & u_L(t) \end{bmatrix}^T$ can be represented as $u(t) = Rr(t)$ where, $r(t) = \begin{bmatrix} r_1 & \dots & r_K \end{bmatrix}^T$ and R is the matrix of transfer function R_{jk}.

(1) represents output of each node as an influence of multiple nodes acting as input in the network. Collecting the equations for all the nodes in the system results in

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1L} \\ G_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G_{L-1}L \\ G_{L1} & \cdots G_{LL-1} & & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + R \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{bmatrix} + H \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_L \end{bmatrix} \tag{2}$$

which in the matrix form is,

$$w = Gw + Rr + He \tag{3}$$

According to the local direct method, *external variable* $r = 0$ can be considered, without loss of generality, thereby simplifying the equation. Furthermore, the local direct method transforms (2) into an estimation problem where we identify the dynamics from inputs to outputs. The equation is given in (4).

$$\underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_o \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} \bar{G}_{\mathcal{Q}\mathcal{Q}} & \bar{G}_{\mathcal{Q}\mathcal{U}} \\ \bar{G}_{o\mathcal{Q}} & \bar{G}_{o\mathcal{U}} \end{bmatrix}}_{\bar{G}} \underbrace{\begin{bmatrix} w_{\mathcal{Q}} \\ w_{\mathcal{U}} \end{bmatrix}}_{w_{\mathcal{D}}} + \underbrace{\begin{bmatrix} \bar{H}_{\mathcal{Q}\mathcal{Q}} & \bar{H}_{\mathcal{Q}\mathcal{U}} \\ \bar{H}_{o\mathcal{Q}} & \bar{H}_{o\mathcal{U}} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} \xi_{\mathcal{Q}} \\ \xi_o \end{bmatrix}}_{\xi_{\mathcal{Y}}} \tag{4}$$

The signals $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ are signals that are taken as outputs and inputs respectively (refer to [18] for defining the sets $\mathcal{Y} = \mathcal{Q} \cup \{o\}$ and $\mathcal{D} = \mathcal{Q} \cup \mathcal{U}$). The sets $\mathcal{Y}$ and $\mathcal{D}$ are defined to obtain a consistent estimate for all the modules in $\bar{G}$. $\mathcal{Q}$ is the set of nodes that are common to both inputs and outputs, $\mathcal{U}$ is the set of nodes that are exclusively inputs and o is the set of nodes that are exclusively outputs. The vector $\xi_{\mathcal{Y}}$ is a white noise signal constructed by spectral decomposition to ensure $\bar{H}$ is monic, minimum phase and stably invertible. The following example illustrates the choice of input and output.

Example : Consider the simple network provided in figure 1. The desired module that needs to be estimated is G_{21}. Therefore, the sets $\mathcal{D} = \{1, 3, 4\}$ and $\mathcal{Y} = \{2\}$. However, since there is correlation between signal at nodes 2 and 4 due to noise, node 4 is also considered as an output node.

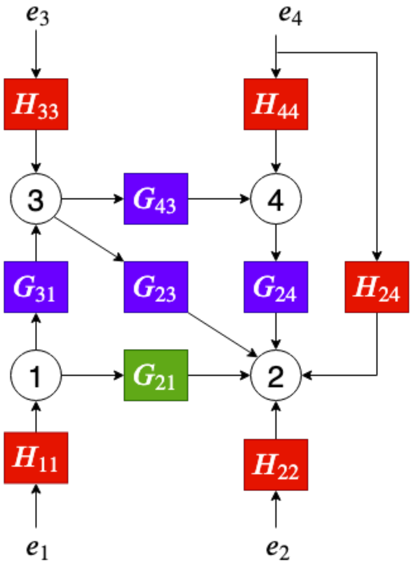

Fig. 1. A simple 4 node network, G_{21} is the target module, and process noise at nodes 2 and 4 are correlated

Therefore, set $\mathcal{Y} = \{2, 4\}$. There are different choices of inputs and outputs that lead to consistent estimates, however, in this case, we choose the algorithm that uses maximum number of inputs and outputs to highlight the challenges in estimation. This results in the following equation,

$$\underbrace{\begin{bmatrix} w_4 \\ w_2 \end{bmatrix}}_{w_{\mathcal{Y}}} = \underbrace{\begin{bmatrix} 0 & 0 & G_{43} \\ G_{24} & G_{21} & G_{23} \end{bmatrix}}_{\bar{G}} \underbrace{\begin{bmatrix} w_4 \\ w_1 \\ w_3 \end{bmatrix}}_{w_{\mathcal{D}}} + \underbrace{\begin{bmatrix} H_{44} & 0 \\ H_{24} & H_{22} \end{bmatrix}}_{\bar{H}} \underbrace{\begin{bmatrix} e_4 \\ e_2 \end{bmatrix}}_{e}. \tag{5}$$

Although the nodal signals $w_{\mathcal{Y}}(t)$ and $w_{\mathcal{D}}(t)$ are time dependent, the dependency is kept tacit for simplicity of notations. Since the noise model $\bar{H}$ should be monic, a spectral decomposition is carried out and $\bar{H}e$ is replaced with $\tilde{H}\xi$ with ξ having a covariance of Λ and $\tilde{H}$ being monic. Now, the modules, G_{23}, G_{24} and G_{43} are to be estimated along with G_{21} for consistency. This results in an estimation problem in which suitable model order needs to be chosen for 4 modules. For the 4 modules, one has to try several possible combinations of orders for each module depending on the model structure. In addition, each module needs to be parameterized according to their corresponding model order, increasing the total number of parameters to be estimated. This increase in the number of parameters increase the variance of the estimates. Therefore, an algorithm needs to be developed to tackle model order selection and reduce the number of parameters.

To develop the algorithm, we assume that N measurements of the signals $w_{\mathcal{Y}}$ and $w_{\mathcal{D}}$ are collected. From these measurements, we are focusing on building a model of the module linking the node i to node j i.e. $G^0_{ji}(q)$, directly. For this purpose, we choose the parametrization of $G^0_{ji}(q)$, denoted by $G_{ji}(q, \theta)$, that describes the dynamics of the module for a certain parameter vector $\theta_0 \in \mathbb{R}^{n_\theta}$.

III. Forming the Empirical Bayes Framework

In this section, we explain the method developed for identifying the module $G^0_{ji}(q)$.

A. The local direct method

Following (4), the dynamic network representation is modified such that the module of interest, $G_{ji}(q, \theta)$ becomes the first element of $\bar{G}$ without affecting the monicity of $\bar{H}$. As per this representation,

$$\underbrace{\begin{bmatrix} w_j \\ w_{\tilde{\mathcal{Y}}} \end{bmatrix}}_{\tilde{w}_{\mathcal{Y}}(t)} = \underbrace{\begin{bmatrix} G_{ji} & \bar{G}_{j\tilde{\mathcal{D}}} \\ \bar{G}_{\tilde{\mathcal{Y}}i} & \bar{G}_{\tilde{\mathcal{Y}}\tilde{\mathcal{D}}} \end{bmatrix}}_{\underline{G}(q,\theta)} \underbrace{\begin{bmatrix} w_i \\ w_{\tilde{\mathcal{D}}} \end{bmatrix}}_{\tilde{w}_{\mathcal{D}}(t)} + \underbrace{\begin{bmatrix} \bar{H}_{jj} & \bar{H}_{j\tilde{\mathcal{Y}}} \\ \bar{H}_{\tilde{\mathcal{Y}}j} & \bar{H}_{\tilde{\mathcal{Y}}\tilde{\mathcal{Y}}} \end{bmatrix}}_{\underline{H}(q,\theta)} \underbrace{\begin{bmatrix} \xi_j \\ \xi_{\tilde{\mathcal{Y}}} \end{bmatrix}}_{\tilde{\xi}_{\mathcal{Y}}(t)} \tag{6}$$

where, $\tilde{\mathcal{Y}} = \mathcal{Y} \backslash \{j\}$, and $\tilde{\mathcal{D}} = \mathcal{D} \backslash \{i\}$. The signals $\tilde{w}_{\mathcal{Y}}$, $\tilde{w}_{\mathcal{D}}$, and $\tilde{\xi}_{\mathcal{Y}}$ contain the same signals as $w_{\mathcal{Y}}$, $w_{\mathcal{D}}$, and $\xi_{\mathcal{Y}}$ respectively, in different orders, suited for (6). If $\underline{H}$ is monic, stable and inversely stable, then, $\bar{H}_{jj}$ is a monic, stable and inversely stable transfer function, $\bar{H}_{\tilde{\mathcal{Y}}\tilde{\mathcal{Y}}}$ is a monic, stable and inversely stable transfer function matrix, and $\bar{H}_{j\tilde{\mathcal{Y}}}$ and $\bar{H}_{\tilde{\mathcal{Y}}j}$ are strictly proper transfer function matrices. From (6), we construct the one-step ahead predictor [11] for $w_{\mathcal{Y}}(t)$,

$$\hat{w}_{\mathcal{Y}}(t|t-1;\theta) = \underline{H}(q)^{-1}\underline{G}(q)w_{\mathcal{D}} + \left(I - \underline{H}(q)^{-1}\right) w_{\mathcal{Y}} \\ + \xi_{\mathcal{Y}} \tag{7}$$

which is a function of parameters of the module $G_{ji}(q, \theta)$ in $\underline{G}(q, \theta)$. In addition to $G_{ji}(q, \theta)$, all additional modules in $\underline{G}(q, \theta)$ and $\underline{H}(q, \theta)$ are suitably parameterized with additional parameters. These parameters are then identified by minimizing the power of prediction error, $\varepsilon(t, \theta) = \underline{H}^{-1}(q, \theta)(w_{\mathcal{Y}}(t) - \underline{G}(q, \theta))$. Since this is a MIMO estimation problem, the number of parameters increase as the number of modules in $\underline{G}(q, \theta)$ and $\underline{H}(q, \theta)$ increase, leading to a complex estimation problem.

B. The Bayesian model

In this section, we discuss how to avoid parameterization of additional modules in (7) using regularized kernel-based methods. The approach followed here is an extension of the MISO approach proposed in [17] for the MIMO estimation problem. For this approach, let us first define the following quantities

$$S(q) = I - \underline{H}(q)^{-1}, \qquad \tilde{G} = \begin{bmatrix} 0 & \bar{G}_{j\tilde{\mathcal{D}}} \\ \bar{G}_{\tilde{\mathcal{Y}}i} & \bar{G}_{\tilde{\mathcal{Y}}\tilde{\mathcal{D}}} \end{bmatrix}$$

$S(q)$ is a $N_{\mathcal{Y}} \times N_{\mathcal{Y}}$ transfer function matrix, where $N_{\mathcal{Y}}$ is the number of nodes in the set $\mathcal{Y}$. $\tilde{G}$ is the matrix of all additional modules except the target module in $\bar{G}$. With these definitions, building a predictor from (6) results in,

$$w_{\mathcal{Y}} = (I - S(q)) \begin{bmatrix} G_{ji}(q, \theta) \\ \mathbf{0}_{(N_{\mathcal{Y}}-1)\times 1} \end{bmatrix} w_i + (I - S(q))\tilde{G}(q)w_{\mathcal{D}} \\ + S(q)w_{\mathcal{Y}} + \xi_{\mathcal{Y}} \tag{8}$$

In order to represent the modules that are not parameterized, we use impulse response models. However, we need to construct impulse response models only for dynamics present in the network. To this end, we need to identify the number of modules in $(I-S(q))\tilde{G}(q)$. The sparsity of $\tilde{G}(q)$ (which entries of $\tilde{G}(q)=0$ after the transformation in (6)) is assumed to be known. With that, we construct two matrices representing the structure of $(I-S(q))$ and $\tilde{G}(q)$,

$$T_S = \begin{bmatrix} 1 & d & \cdots & d \\ d & 1 & \cdots & d \\ \vdots & \vdots & \ddots & \vdots \\ d & d & \cdots & 1 \end{bmatrix}, \quad T_{\tilde{G}}(m,n) = \begin{cases} 0 & \text{if } \tilde{G}(m,n)=0 \\ 1 & \text{otherwise} \end{cases}$$

$$T_{S\tilde{G}} = T_S \times T_{\tilde{G}}$$

where, d and 1 represents the presence and absence of delay in the transfer functions respectively in $(I-S(q))$. Each non-zero entity in $T_{S\tilde{G}}$ represents a direct causal relation between the corresponding input and output nodes. This causal relation is denoted as $S_p \ \forall \ p \in \Omega_{S\tilde{G}}$, where $\Omega_{S\tilde{G}} = \{(a,b), \ a \in \mathcal{Y}, \ b \in \mathcal{D} \mid T_{S\tilde{G}}(a,b) \neq 0\}$. The impulse response of S_p is represented as $S_p = \sum_{n=1}^{\infty} s_p(d)q^{-d} \ \forall \ p$ which is an infinite sum. Therefore, the first ℓ coefficients of the impulse response of $S_p \ \forall \ p$ is stacked in the vector $s_{\mathcal{D}} \in \mathbb{R}^{\ell N_{\Omega_{S\tilde{G}}} \times 1}$, where $N_{\Omega_{S\tilde{G}}}$ is the number of elements in the set $\Omega_{S\tilde{G}}$. The integer ℓ is chosen such that $s_p(\ell+1) \approx 0$. Similarly, $s_{\mathcal{Y}} \in \mathbb{R}^{\ell N_{\Omega_{\mathcal{YY}}} \times 1}$ is generated for all the modules $S_k \forall \ k \in \Omega_{\mathcal{YY}}$ where, $\Omega_{\mathcal{YY}} = \{(a,b), \ a,b \in \mathcal{Y}\}$, and $N_{\Omega_{\mathcal{YY}}} = N_{\mathcal{Y}} N_{\mathcal{Y}}$ is the number of elements in the set $\Omega_{\mathcal{YY}}$. S_k denotes the impulse response models of the transfer functions in $S^0(q)$. Now we define the vector notation for the output node signals,

$$w_j = \begin{bmatrix} w_j(1) & \ldots & w_j(N) \end{bmatrix}^\top,$$

$$w_n = \begin{bmatrix} w_n(1) & \ldots & w_n(N) \end{bmatrix}^\top, \ \forall \ n \in \tilde{\mathcal{Y}},$$

$$w_{\mathcal{Y}} = \begin{bmatrix} w_j^\top & w_{n1}^\top & \ldots & w_{n*}^\top \end{bmatrix}^\top,$$

where, $n1, \ \ldots, \ n*$ are the elements of the set $\tilde{\mathcal{Y}}$,

$$\tilde{w}_k = \begin{bmatrix} 0 & w_k(1) & \ldots & w_k(N-1) \end{bmatrix}^\top \ \forall \ k \in \mathcal{Y},$$

and the vector notation for the input node signals,

$$\tilde{w}_m = \begin{bmatrix} 0 & w_m(1) & \ldots & w_m(N-1) \end{bmatrix}^\top \ \forall \ m \in \mathcal{D}.$$

From the vector notations, we define the matrices, $W_m \in \mathbb{R}^{N\times\ell}$ as the toeplitz of $\tilde{w}_m$, and $W_k \in \mathbb{R}^{N\times\ell}$ as the toeplitz of $\tilde{w}_k$. From the above matrices definition, we define the matrices,

$$W_{\mathcal{Y}} = \begin{bmatrix} W_{k1} & \ldots & W_{k*} \end{bmatrix},$$

where, $k1, \ \ldots, \ k*$ are the elements of the set $\mathcal{Y}$,

$$W_{\mathcal{D}u} = \begin{bmatrix} W_{m1} & \ldots & W_{m*} \end{bmatrix},$$

where, $u \in \mathcal{Y}$, and $m1, \ \ldots, \ m*$ are the input nodes that have a direct causal connection to the u^{th} output. This set of input nodes are obtained from the non-zero entries in $T_{S\tilde{G}}$ that correspond to the u^{th} output.

From the definition of matrices, $W_{\mathcal{D}_u}$ and $W_{\mathcal{Y}}$, the matrices

$$\boldsymbol{W}_{\mathcal{D}} = \begin{bmatrix} W_{\mathcal{D}u1} & & \\ & \ddots & \\ & & W_{\mathcal{D}u*} \end{bmatrix}_{NN_{\mathcal{Y}} \times \ell N_{\Omega_{S\tilde{G}}}}$$

$$\boldsymbol{W}_{\mathcal{Y}} = \begin{bmatrix} W_{\mathcal{Y}} & & \\ & \ddots & \\ & & W_{\mathcal{Y}} \end{bmatrix}_{NN_{\mathcal{Y}} \times \ell N_{\mathcal{Y}} N_{\mathcal{Y}}}$$

are constructed. Here, $u1, \ \ldots, \ u*$ correspond to nodes in the set $\mathcal{Y}$. The above matrices are defined to represent the ℓ length convolution of impulse response models of the additional modules with the inputs. However, the parameterized $G_{ji}(q,\theta)$ should also be represented as an impulse response model and convoluted with its input. Therefore, the impulse response of $G_{ji}(q,\theta)$ is written as $G_{ji}(q,\theta) = \sum_{d=1}^{\infty} g_{ji}(d)q^{-d}$. Then, the first N coefficients of this impulse response model is collected and represented in vector form as $g_{ji} = \begin{bmatrix} g_{ji}(1) & \ldots & g_{ji}(N) \end{bmatrix}^\top$. Though the vector $g_{ji}(\theta)$ is a function of the parameter vector θ, for ease of notation, the dependency is kept implicit. Now, we define $W_i \in \mathbb{R}^{N\times N}$ as the toeplitz of $\begin{bmatrix} 0 & w_i(1) & \ldots & w_i(N-1) \end{bmatrix}^\top$, $G_\theta \in \mathbb{R}^{N\times N}$ as the toeplitz of g_{ji}, and $\bar{W}_i \in \mathbb{R}^{N\times\ell}$ as the toeplitz of $\begin{bmatrix} 0 & 0 & w_i(1) & \ldots & w_i(N-2) \end{bmatrix}^\top$ so that $\tilde{W}_i = [G_\theta \bar{W}_i \ \ \mathbf{0}]$, and

$$\tilde{\boldsymbol{W}}_i = \begin{bmatrix} \tilde{W}_i & & \\ & \ddots & \\ & & \tilde{W}_i \end{bmatrix}_{NN_{\mathcal{Y}} \times \ell N_{\mathcal{Y}} N_{\mathcal{Y}}}$$

are constructed. Once the above matrices are defined, we simplify the notation and represent (8) as follows

$$w_{\mathcal{Y}} = \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} - \tilde{\boldsymbol{W}}_i s_{\mathcal{Y}} + \boldsymbol{W}_{\mathcal{D}} s_{\mathcal{D}} + \boldsymbol{W}_{\mathcal{Y}} s_{\mathcal{Y}} + \xi \qquad (9)$$

where, $\xi \in \mathbb{R}^{NN_{\mathcal{Y}} \times 1}$ represents the stacked vectorized noise for each output. The impulse response coefficients in (9) need to be modeled and since our goal is to limit the number of estimated parameters, we consider a non-parametric model for the impulse responses $s_{\mathcal{Y}}$ and $s_{\mathcal{D}}$. As a result, we use parametric modeling for g_{ji} and non-parametric model induced by Gaussian process for all the impulse responses in $s_{\mathcal{Y}}$ and $s_{\mathcal{D}}$ in (9). Gaussian processes are effective in reducing the variance of the impulse response estimate with suitable choice of a prior covariance matrix [15]. Therefore, we model each of the impulse responses in $s_{\mathcal{Y}}$ and $s_{\mathcal{D}}$ using independent Gaussian processes with zero mean and a suitable covariance matrix. The covariance matrices for these processes are chosen to be that of a *First order Stable Spline kernel* [4]. The kernel structure is given by $K := \lambda K_\beta$ with

$$[K_\beta]_{x,y} = \beta^{\max(x,y)}$$

where, $\beta \in [0,1)$ and $\lambda \geq 0$. λ and β are hyperparameters that govern the amplitude and exponential decay of the Gaussian vector respectively. The Stable spline kernel offers properties of smoothness and stability making it a suitable choice for modelling impulse response. Therefore, each impulse response model in $s_{\mathcal{D}}$ and $s_{\mathcal{Y}}$ are represented using independent Gaussian processes.

$$s_p \sim \mathcal{N}(\mathbf{0}, \lambda_p K_{\beta_p}),\ p \in \Omega_{S\tilde{G}}$$
$$s_k \sim \mathcal{N}(\mathbf{0}, \lambda_k K_{\beta_k}),\ k \in \Omega_{\mathcal{YY}}$$

We assign independent hyperparameters to each impulse response model for flexibility in modelling. Let us now define,

$$\boldsymbol{s} := \begin{bmatrix} s_{\mathcal{Y}}^{\top} & s_{\mathcal{D}}^{\top} \end{bmatrix} \tag{10}$$

$$\mathbf{W} := \begin{bmatrix} \boldsymbol{W}_{\mathcal{Y}} - \tilde{\boldsymbol{W}}_i & \boldsymbol{W}_{\mathcal{D}} \end{bmatrix} \tag{11}$$

and,

$$\mathbf{K} := \operatorname{diag}\{\lambda_{p1} K_{\beta_{p1}}, \ldots \lambda_{p*} K_{\beta_{p*}}, \lambda_{k1} K_{\beta_{k1}}, \ldots \lambda_{k*} K_{\beta_{k*}}\} \tag{12}$$

where, $p1,\ \ldots,\ p*$ are elements of the set $\Omega_{S\tilde{G}}$, and $k1,\ \ldots,\ k*$ are elements of the set $\Omega_{\mathcal{YY}}$. Using the above definitions, (9) can be written as,

$$w_{\mathcal{Y}} = \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} + \mathbf{W}\boldsymbol{s} + \xi \tag{13}$$

In (13), $\boldsymbol{s}$ is modeled as Gaussian process. Therefore if we consider a Gaussian distribution for noise ξ

$$\xi \sim \mathcal{N}(\mathbf{0}, \Sigma)$$
$$\Sigma := \begin{bmatrix} \sigma_{11}^2 I_N & \sigma_{12}^2 I_N & \ldots & \sigma_{1k}^2 I_N \\ * & \sigma_{22}^2 I_N & \ldots & \sigma_{2k}^2 I_N \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \ldots & \sigma_{kk}^2 I_N \end{bmatrix}$$

then, we can write a joint probabilistic description of $\boldsymbol{s}$ and $w_{\mathcal{Y}}$ as a joint Gaussian representation,

$$p\left(\begin{bmatrix} \mathbf{s} \\ w_{\mathcal{Y}} \end{bmatrix}; \eta\right) \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}\mathbf{W}^{\top} \\ \mathbf{W}\mathbf{K} & \mathbf{P} \end{bmatrix}\right) \tag{14}$$

where,

$$\mathbf{P} = \Sigma + \mathbf{W}\mathbf{K}\mathbf{W}^{\top}$$

and,

$$\begin{aligned} \eta = [&\theta \quad \lambda_{p1} \quad \ldots \quad \lambda_{p*} \quad \lambda_{k1} \quad \ldots \quad \lambda_{k*} \quad \beta_{p1} \quad \ldots \\ &\beta_{p*} \quad \beta_{k1} \quad \ldots \quad \beta_{k*} \quad \sigma_{11}^2 \quad \ldots \quad \sigma_{1k}^2 \quad \sigma_{22}^2 \\ &\ldots \quad \sigma_{2k}^2 \quad \ldots \quad \sigma_{kk}^2]. \end{aligned}$$

The vector η is a vector of parameters that drive the probability distribution function in (14). It consists of the parameters of $G_{ji}(\theta)$, the hyperparameters of the impulse response models and the covariance of the noise acting on $w_{\mathcal{Y}}$. Since the goal is to estimate the θ contained in η, we need to estimate η based on the data $w_{\mathcal{Y}}$. As a result, we adopt an Empirical Bayes (EB) framework [12] which estimates the prior based on the data. To this end, we consider the marginal pdf of $w_{\mathcal{Y}}$ by integrating out the effect of $\boldsymbol{s}$ and maximize the likelihood of η. The objective function therefore becomes,

$$\begin{aligned} \hat{\eta} &= \operatorname*{argmax}_{\eta} p(w_{\mathcal{Y}}; \eta) \\ &= \operatorname*{argmin}_{\eta} \log|\mathbf{P}| + \left(w_{\mathcal{Y}} - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}\right)^{\top} \mathbf{P}^{-1} \left(w_{\mathcal{Y}} - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}\right) \end{aligned} \tag{15}$$

The above optimization problem is complex and nonlinear, and solving the problem would be cumbersome. In the next section, we introduce a method to solve the marginal likelihood maximization problem.

IV. Maximizing Marginal Likelihood

To obtain the solution to the marginal likelihood maximization, we consider the *Expectation Maximization* (EM) method [7]. The EM algorithm is a suitable method to solve (15) since it guarantees convergence to a stationary point [3]. Also, by suitable choice of posterior, the optimization problem is simplified, as will be seen in *Lemma 1*. In this algorithm, we need a *latent variable* which is unobservable and simplifies the computation of the marginal likelihood. Therefore $\boldsymbol{s}$ is chosen as the latent variable in this case. The algorithm consists of two steps, namely the *Expectation* step and the *Maximization* step which are explained below.

- *Expectation step*: Given an estimate of $\hat{\eta}^{(n)}$ computed at the n^{th} iteration, compute the expected value of the joint log-likelihood of $w_{\mathcal{Y}}$ and $\boldsymbol{s}$ with respect to the posterior $p(\boldsymbol{s}|w_{\mathcal{Y}}; \hat{\eta}^{(n)})$
$$Q^{(n)}(\eta) = \mathbb{E}[\log p(w_{\mathcal{Y}}, \boldsymbol{s}; \eta)], \tag{16}$$
- *Maximization step*: Update $\hat{\eta}$ by solving the following problem
$$\hat{\eta}^{(n+1)} = \operatorname*{argmax}_{\eta} Q^{(n)}(\eta) \tag{17}$$

The estimate is obtained by iterating between (16) and (17). Although the computation is recurring, if the solution through the algorithm is fast and computationally less complex than (15) then it proves beneficial. In the following subsections, We explore if this is indeed true.

A. Computation of Expectation-step

The Expectation is taken with respect to the posterior $p(\boldsymbol{s}|w_{\mathcal{Y}}; \eta)$. Therefore, the posterior distribution of $\boldsymbol{s}$ given $w_{\mathcal{Y}}$ and an estimate of η is given by the following equation [1],

$$p(\boldsymbol{s}|w_{\mathcal{Y}}; \eta) \sim \mathcal{N}(\boldsymbol{s}_m, P_{\boldsymbol{s}}) \tag{18}$$

where,

$$\begin{aligned} P_{\boldsymbol{s}} &= \mathbf{K} - \mathbf{K}\mathbf{W}^{\top}(\mathbf{W}\mathbf{K}\mathbf{W}^{\top} + \Sigma)^{-1}\mathbf{W}\mathbf{K}, \\ \boldsymbol{s}_m &= (\mathbf{K}\mathbf{W}^{\top}(\mathbf{W}\mathbf{K}\mathbf{W}^{\top} + \Sigma)^{-1})(w_{\mathcal{Y}} - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji}) \end{aligned} \tag{19}$$

Let $\hat{\boldsymbol{s}}^{(n)}$ and $\hat{P}_{\boldsymbol{s}}^{(n)}$ be the posterior mean and covariance of $\boldsymbol{s}$ obtained from (19) using $\hat{\eta}^{(n)}$. Let us define $\hat{\boldsymbol{S}}^{(n)} := \hat{P}_{\boldsymbol{s}}^{(n)} + \hat{\boldsymbol{s}}^{(n)}\hat{\boldsymbol{s}}^{(n)\top}$ as the posterior second moments of $\hat{\boldsymbol{s}}$ and consider its $\ell \times \ell$ diagonal blocks and denote them as $\hat{\boldsymbol{S}}_m^{(n)}$ $\forall\ m$ where, $m \in \Omega_{S\tilde{G}} \cup \Omega_{\mathcal{YY}}$. These blocks correspond to the posterior second moments of the corresponding impulse response coefficients.
The structure of $Q^{(n)}(\eta)$ for the problem considered in (15) is defined in the following lemma.

Lemma 1. *Let $\hat{\eta}^{(n)}$ be the estimate of η at n^{th} iteration of the EM algorithm, then*

$$Q^{(n)}(\eta) = Q_0^{(n)}(\theta, \varsigma) + \sum_{m \,\in\, \Omega_{S\tilde{G}} \cup \Omega_{\mathcal{Y}\mathcal{Y}}} Q_{s_m}^{(n)}(\lambda_m, \beta_m) \quad (20)$$

where, $\varsigma = \begin{bmatrix}\sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{kk}^2\end{bmatrix}$ is the vector of noise covariance that constitute Σ,

$$\begin{aligned} Q_0^{(n)}(\theta, \varsigma) = -\log\det\Sigma - tr\Bigg(\Sigma^{-1}\Bigg(& w_{\mathcal{Y}} w_{\mathcal{Y}}^\top \\ + \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} + \mathbf{W}\hat{\boldsymbol{S}}^{(n)}\mathbf{W}^\top - \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} w_{\mathcal{Y}}^\top \\ - \mathbf{W}\hat{\boldsymbol{s}}^{(n)} w_{\mathcal{Y}}^\top - w_{\mathcal{Y}} g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} + \mathbf{W}\hat{\boldsymbol{s}}^{(n)} g_{ji}^\top \begin{bmatrix} W_i^\top & \mathbf{0}^\top \end{bmatrix} \\ - w_{\mathcal{Y}} \hat{\boldsymbol{s}}^{(n)\top}\mathbf{W}^\top + \begin{bmatrix} W_i \\ \mathbf{0} \end{bmatrix} g_{ji} \hat{\boldsymbol{s}}^{(n)\top}\mathbf{W}^\top \Bigg)\Bigg) \end{aligned}$$

and,

$$Q_{s_m}^{(n)}(\lambda_m, \beta_m) = -\log\det\lambda_m K_{\beta_m} - tr\Big((\lambda_m K_{\beta_m})^{-1}\hat{\boldsymbol{S}}_m^{(n)}\Big).$$

Proof. For proof, see appendix A. □

(20) is seen as a sum of expectations that depends on different parameters in η. As a result, it can be concluded that the complex optimization problem of (15) is simplified into smaller optimization problems, which can all be solved independently of each other. Therefore the maximization step splits into two parts, one for updating the hyperparameters λ_m and β_m, and one for updating θ and Σ. The updation of η is discussed in the following subsection.

B. Computation of Maximization-step

From (20) it is evident that the hyperparameters of the kernel can be updated independent of the remaining parameters. Therefore, *Theorem* 1 gives the steps for updating the hyperparameters, β_m and λ_m.

Theorem 1. *Define,*

$$Q_{\beta_m}^{(n)}(\beta_m) = \ell \log\left(tr(K_{\beta_m}^{-1}\hat{S}_m^{(n)})\right) + \log\det K_{\beta_m} \quad (21)$$

for $m \in \Omega_{S\tilde{G}} \cup \Omega_{\mathcal{Y}\mathcal{Y}}$. Then,

$$\begin{aligned} \hat{\beta}_m^{(n+1)} &= \underset{\beta_n \in [0,1)}{\operatorname{argmin}}\, Q_{\mathcal{B}}^{(n)}(\beta_m) \\ \hat{\lambda}_m^{(n+1)} &= \frac{1}{\ell} tr(K_{\beta_m^{(n+1)}}^{-1}\hat{S}_m^{(n)}) \end{aligned} \quad (22)$$

Proof. For proof, see appendix B. □

The optimization problem in (21) is a scalar optimization problem and is computationally fast [2] [9]. Once an estimate of $\hat{\beta}_m^{(n+1)}$ is obtained, $K_{\beta_m^{(n+1)}}$ is computed. The computed value is used for updating $\hat{\lambda}_m^{(n+1)}$. The equation for updating $\hat{\lambda}_m^{(n+1)}$ is a closed form solution, requiring no optimization. Therefore the updation of hyperparameters λ_m and β_m require minimal effort. Once an estimate is obtained, the posterior and second moment of the posterior are calculated from (19). We now turn our attention towards optimization of θ and Σ. Updating θ and Σ is defined by *Theorem 2*.

Theorem 2. *The estimates, $\hat{\theta}^{(n+1)}$ and $\hat{\Sigma}^{(n+1)}$ is obtained from,*

$$\begin{bmatrix} \hat{\theta}^{(n+1)} \\ \hat{\varsigma}^{(n+1)} \end{bmatrix} = \underset{\theta,\varsigma}{\operatorname{argmin}} -Q_0^{(n)}(\theta, \varsigma) \quad (23)$$

and $\hat{\Sigma}^{(n+1)}$ is obtained from $\varsigma^{\widehat{(n+1)}}$.

Proof. The objective function in (20) that is dependent only on θ and Σ is used for optimizing. The maximization is converted to minimization due to sign inversion. □

Although there exists an objective function that can be optimized to obtain an estimate of θ and Σ, solving it is a cumbersome procedure. Additionally, the complexity of the solution increases with the number of outputs as the size of Σ increases with the number of outputs. Therefore, there is a necessity to simplify the optimization problem stated in (23) such that, the output dependent Σ does not increase the complexity of optimization for θ.

V. Discussion and Further work

In the optimization of θ and $\Sigma(\varsigma)$, the objective function is non-linear in both the parameters. Due to ς, the optimization problem becomes a problem with increasing complexity. While size of θ does not increase with increase in predicted outputs, the size of ς increases. This makes the optimization problem not scalable. To resolve the scalabilty, the optimal value of ς should be updated using a closed form solution [20]. With a closed form solution for updating the estimate of ς, only θ will be updated using a non-linear objective function. Hence, using the simplified optimization of θ and Σ, a scalable algorithm can be developed.

Performance of the scalable algorithm should be tested on a sample network. The results obtained could be compared with existing methods for local module identification such as direct method [18] and Empirical Bayes Direct method (EBDM) [17]. Thereby, the variance and bias of the estimated parameters can be compared and the developed algorithm can be evaluated.

Additionally, during the development of the algorithm, it is assumed that the structure of $\tilde{G}$ (in Sec.III-B) is known. With this assumption, only the necessary and existing modules are modelled. However, in complex networks, this information is hardly present. In addition, all entries of the noise model $\underline{H}$ are modelled in this study. This is, however, not necessary in the example network depicted in figure 1. In this example, the noise model takes the form of a diagonal transfer function matrix after spectral decomposition. The aforementioned issues appear due to presence of sparsity in the dynamic network. It is therefore, interesting to explore identification of the target module in a sparse dynamic network.

VI. Conclusion

In this report, an approach towards learning dynamic networks with correlated noise using regularized kernel-based methods is presented. An EB method for the MIMO identification problem of dynamic networks is developed similar to EBDM which is applicable only in MISO identification problem. In this method, the desired module is represented as a parametric model and the additional modules are represented as Gaussian processes governed by a stable spline kernel and circumvents the need to select a suitable model order for non-target modules. In addition, by using regularized methods, the number of parameters that need to be estimated is also greatly reduced. Although the estimation problem has been simplified by reducing the number of parameters, the complexity of the problem still depends on the number of modules that need to be estimated. In other terms, the algorithm is not scalable. Therefore, further steps need to be taken to make the algorithm implementable.

Appendix

A. *Proof of Lemma 1*

Following Bayes' theorem, (16) can be written as follows,

$$Q^{(n)}(\eta) = \mathbb{E}[\log p(w_{\mathcal{Y}}|\boldsymbol{s};\eta)] + \mathbb{E}[\log p(\boldsymbol{s};\eta)] \tag{24}$$

Define

$$\begin{aligned}\mathcal{A} = & -\frac{N\mathcal{Y}}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma \\ & -\frac{1}{2}\left(w_y - \begin{bmatrix} W_i \\ \mathbf{0}\end{bmatrix} g_{ji} - \mathbf{W}\boldsymbol{s}\right)^T \Sigma^{-1}\left(w_y - \begin{bmatrix} W_i \\ \mathbf{0}\end{bmatrix} g_{ji} - \mathbf{W}\boldsymbol{s}\right)\end{aligned} \tag{25}$$

and

$$\begin{aligned}\mathcal{B} = \sum_{m \,\in\, \Omega_{S\tilde{G}} \cup \Omega_{\mathcal{Y}\mathcal{Y}}} \Bigg(& -\frac{l}{2}\log(2\pi) - \frac{1}{2}\log\det\lambda_m K_{\beta_m} \\ & - \frac{1}{2} s_m^T(\lambda_m K_{\beta_m})^{-1} s_m \Bigg)\end{aligned} \tag{26}$$

Using properties of trace, removing constant terms, multiplying by 2 and taking the expectation with respect to posterior, (25) and (26) are written as,

$$\begin{aligned}\mathbb{E}[\mathcal{A}] = & -\log\det\Sigma - \text{tr}\Bigg(\Sigma^{-1}\Bigg(w_y w_y^T \\ & + \begin{bmatrix} W_i \\ \mathbf{0}\end{bmatrix} g_{ji} g_{ji}^T \begin{bmatrix} W_i^T & \mathbf{0}^T\end{bmatrix} + \mathbf{W}\boldsymbol{s}\boldsymbol{s}^T\mathbf{W}^T - \begin{bmatrix} W_i \\ \mathbf{0}\end{bmatrix} g_{ji} w_y^T \\ & - \mathbf{W}\boldsymbol{s} w_y^T - w_y g_{ji}^T \begin{bmatrix} W_i^T & \mathbf{0}^T\end{bmatrix} + \mathbf{W}\boldsymbol{s} g_{ji}^T \begin{bmatrix} W_i^T & \mathbf{0}^T\end{bmatrix} \\ & - w_y \boldsymbol{s}^T\mathbf{W}^T + \begin{bmatrix} W_i \\ \mathbf{0}\end{bmatrix} g_{ji}\boldsymbol{s}^T\mathbf{W}^T\Bigg)\Bigg)\end{aligned} \tag{27}$$

$$\mathbb{E}[\mathcal{B}] = \sum_m -\log\det\lambda_m K_{\beta_m} - \text{tr}\left((\lambda_m K_{\beta_m})^{-1} s_m s_m^T\right) \tag{28}$$

By substituting, $\boldsymbol{s}$ as $\hat{\boldsymbol{s}}^{(n)}$, $\boldsymbol{s}\boldsymbol{s}^T$ as $\hat{\boldsymbol{S}}^{(n)}$, s_m as $\hat{s}_m^{(n)}$ and $s_m s_m^T$ as $\hat{S}_m^{(n)}$ in (27) and (28), (20) is obtained.

B. *Proof of Theorem 1*

We consider $Q_{s_m}(\lambda_m, \beta_m)$ in (20) and differentiate it with respect to λ_m. The derivative is then equated to 0 to obtain the expression for λ_m

$$\lambda_m = \frac{1}{\ell}\text{tr}(K_{\beta_m})^{-1}\hat{S}_m \tag{29}$$

(29) is then substituted in the $Q_{s_m}(\lambda_m, \beta_m)$ to eliminate λ_m, and with the change of sign, resulting in the following equation.

$$Q_B(\beta_m) = \ell\log\text{tr}(K_{\beta_m}^{-1}\hat{S}_m) + \log\det K_{\beta_m} \tag{30}$$

Once β_m is obtained, then (29) is used to obtain λ_m z

References

[1] B. D. O. Anderson, J. B. Moore, and M. Eslami. Optimal filtering. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(2):235–236, March 1982.

[2] G. Bottegal, A. Y. Aravkin, H. Hjalmarsson, and G. Pillonetto. Robust EM kernel-based methods for linear system identification. *Automatica*, 67:114 – 126, 2016.

[3] A. R. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(1):47–50, 1983.

[4] T. Chen, H. Ohlsson, and L. Ljung. On the estimation of transfer functions, regularizations and gaussian processes—revisited. *Automatica*, 48(8):1525 – 1535, 2012.

[5] A. Chiuso and G. Pillonetto. A bayesian approach to sparse dynamic network identification. *Automatica*, 48(8):1553 – 1565, 2012.

[6] A. Dankers, P. M. J. Van den Hof, D. Materassi, and H. H. M. Weerts. Conditions for handling confounding variables in dynamic networks. *IFAC-PapersOnLine*, 50(1):3983 – 3988, 2017. 20th IFAC World Congress.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] P. M. J. Van den Hof, A.G. Dankers, P.S.C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods - basic methods for consistent module estimates,. *Automatica*, 49(10):2994–3006, 2013.

[9] N. Everitt, G. Bottegal, and H. Hjalmarsson. An empirical bayes approach to identification of modules in dynamic networks. *Automatica*, 91:144 – 151, 2018.

[10] A. Haber and M. Verhaegen. Subspace identification of large-scale interconnected systems. *IEEE Transactions on Automatic Control*, 59(10):2754–2759, Oct 2014.

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

[12] J. S. Maritz and T. Lwin. *Empirical Bayes Methods*. Routledge, 1989.

[13] D. Materassi and G. Innocenti. Topological identification in networks of dynamical systems. *IEEE Transactions on Automatic Control*, 55(8):1860–1871, Aug 2010.

[14] S. Nabavi and A. Chakrabortty. Topology identification for dynamic equivalent models of large power system networks. In *2013 American Control Conference*, pages 1138–1143, June 2013.

[15] 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*, 50(3):657 – 682, 2014.

[16] G. Prando, A. Chiuso, and G. Pillonetto. Maximum entropy vector kernels for mimo system identification. *Automatica*, 79:326 – 339, 2017.

[17] K. R. Ramaswamy, G. Bottegal, and P. M. J. Van den Hof. Local module identification in dynamic networks using regularized kernel-based methods. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4713–4718, Dec 2018.

[18] K. R. Ramaswamy and P. M. J. Van den Hof. A local direct method for module identification in dynamic networks with correlated noise, 2019. Submitted for publication in *IEEE Trans. Automatic Control*.

[19] 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*, 98:256 – 268, 2018.

[20] K.J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5):551 – 574, 1980.