An empirical Bayes approach to identification of modules in dynamic networks

Niklas Everitt [a], Giulio Bottegal [a], Håkan Hjalmarsson [a]

[a] *ACCESS Linneaus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden*

Abstract

We present a new method of identifying a specific module in a dynamic network, possibly with feedback loops. Assuming known topology, we express the dynamics by an acyclic network composed of two blocks where the first block accounts for the relation between the known reference signals and the input to the target module, while the second block contains the target module. Using an empirical Bayes approach, we model the first block as a Gaussian vector with covariance matrix (kernel) given by the recently introduced stable spline kernel. The parameters of the target module are estimated by solving a marginal likelihood problem with a novel iterative scheme based on the Expectation-Maximization algorithm. Additionally, we extend the method to include additional measurements downstream of the target module. Using Markov Chain Monte Carlo techniques, it is shown that the same iterative scheme can solve also this formulation. Numerical experiments illustrate the effectiveness of the proposed methods.

Key words: system identification, dynamic network, empirical Bayes, expectation-maximization.

1 Introduction

Networks of dynamical systems are everywhere, and applications are in different branches of science, e.g., econometrics, systems biology, social science, and power systems. Identification of these networks, usually referred to as *dynamic networks*, has been given increasing attention in the system identification community, see e.g., Materassi and Innocenti (2010), Van den Hof et al. (2013), Hjalmarsson (2009).

In this paper, we use dynamic network to mean the interconnection of *modules*, where each module is a linear time-invariant (LTI) system. The interconnecting signals are the outputs of these modules. In a graph interpretation, the interconnecting signals represent nodes and the modules represent the edges of the graph. Moreover, we assume that exogenous measurable signals may affect the dynamics of the network.

Two main problems arise in dynamic network identification. The first is unraveling the network topology (i.e., identify the edges of the graph), which can be seen as a model structure selection problem. The second problems is the identification of one or more specific modules in the network.

Some recent papers deal with both the aforementioned problems (Materassi and Salapaka; 2012; Chiuso and Pillonetto; 2012; Materassi and Innocenti; 2010; Hayden et al.; 2014), whereas others are mainly focused on the identification of a single module in the network (Dankers et al.; 2013; Gunes et al.; 2014; Dankers et al.; 2015; Haber and Verhaegen; 2014; Torres et al.; 2014). In particular, Dankers et al. (2013), and Van den Hof et al. (2013) study the problem of understanding which of the available output measurements should be used to obtain a consistent estimate of a target module. In Dankers et al. (2015) instead, errors-in-variables dynamic networks are considered, and methods that lead to consistent module estimates are proposed. As observed in Van den Hof et al. (2013), dynamic networks with known topology can be seen as a generalization of simple compositions, such as systems in cascade, series or feedback connection. Therefore, identification techniques for dynamic networks may be derived by extending methods already developed for simple structures. This is the idea underlying the method presented

in Van den Hof et al. (2013), which generalizes the two-stage method, originally developed for closed-loop systems, to dynamic networks (Forssell and Ljung; 1999). Instrumental variable methods for closed-loop systems (Gilson and Van den Hof; 2005) are adapted to networks in Dankers et al. (2015). Similarly, the methodology proposed in Wahlberg et al. (2009) for the identification of cascaded systems is generalized to the context of dynamic networks in Gunes et al. (2014). In that work, the underlying idea is that a dynamic network can be transformed into an acyclic structure, where any reference signal of the network is the input to a cascaded system consisting of two LTI blocks. In this alternative system description, the first block captures the relation between the reference and the noisy input of the target module, the second block contains the target module. The two LTI blocks are identified simultaneously using the prediction error method (PEM) (Ljung; 1998). In this setup, determining the model structure of the first block of the cascaded structure may be complicated, due to the possibly large number of interconnections in the dynamic network. Furthermore, it requires knowledge of the model structure of essentially all modules in the feedback loop. Therefore, in Gunes et al. (2014), the first block is modeled by an unstructured finite impulse response (FIR) model of high order. The major drawback of this approach is that, as is usually the case with estimated models of high order, the variance of the estimated FIR model is high. The uncertainty in the estimate of the FIR model of the first block will in turn decrease the accuracy of the estimated target module.

The objective of this paper is to propose a method for the identification of a module in dynamic networks that circumvents the high variance that is due to the high order model of the first block. The main contributions of this paper are two-fold. First, we discuss the case where only the sensors directly measuring the input and the output of the target module are used in the identification process. Following a recent trend in system identification, we use regularization to control the variance (Chen et al.; 2012). In particular, by exploiting the equivalence between regularization and Gaussian process regression (Pillonetto et al.; 2014), we model the impulse response of the first block as a zero-mean stochastic process. The covariance matrix is given by the recently introduced *first-order stable spline kernel* (Pillonetto and De Nicolao; 2010), whose structure is parametrized by two *hyperparameters*. An estimate of the target module is then obtained by empirical Bayes (EB) arguments, that is, by maximization of the marginal likelihood of the available measurements (Pillonetto et al.; 2014). This likelihood depends not only on the parameter of the target module, but also on the kernel hyperparameters and the variance of the measurement noise. Therefore, it is required to estimate all these quantities. This is done by designing a novel iterative solution scheme based on an EM-type algorithm (Dempster et al.; 1977), known

as the Expectation/Conditional-Maximization (ECM) algorithm (Meng and Rubin; 1993), which alternates the so called expectation step (E-step) with a series of conditional-maximization steps (CM-steps). When only the module input and output sensors are used, the E-step admits an analytical expression, because joint likelihood of the module output and the sensitivity function is Gaussian. As for the CM-steps, one has to solve relatively simple optimization problems, which either admit a closed form solution, or can be efficiently solved using gradient descent strategies. Therefore, the overall optimization scheme for solving the marginal likelihood problem turns out computationally efficient.

The second main contribution of the paper deals with the case where more sensors spread in the network are used in the identification of the target module. Adding information through addition of measurements used in the identification process has the potential to further reduce the variance of the estimated module (Everitt et al.; 2017). The downside is that an additional measurement comes with another module to estimate, also increasing the number of parameters to estimate. To keep the number of additional parameters to estimate low, we propose a method that exploits regularization, modeling as a Gaussian process also the impulse response of the path linking the target module to any additional sensor. In this case, however, the measured outputs and the unknown paths do not admit a joint Gaussian description. As a consequence, the E-step of the ECM method does not admit an analytical expression, as opposed to the one-sensor case described above. To overcome this issue, we use Markov Chain Monte Carlo (MCMC) techniques (Gilks et al.; 1995) to solve the integral associated with the E-step. In particular, we design an integration scheme based on the Gibbs sampler (Geman and Geman; 1984) that, in combination with the ECM method, builds up a novel identification method for the target module reminiscent of the so called empirical Bayes Gibbs sampling (Casella; 2001).

The effectiveness of the proposed methods is demonstrated through numerical experiments. The methods proposed in this paper are close in spirit to some recently proposed kernel-based techniques for blind system identification (Bottegal et al.; 2015) and Hammerstein system identification (Risuleo et al.; 2015). A part of this paper has previously been presented in Everitt, Bottegal, Rojas and Hjalmarsson (2016). More specifically, the case where only the sensors directly measuring the input and the output of the target module are used in the identification process where partly covered in Everitt, Bottegal, Rojas and Hjalmarsson (2016), whereas, the method where more sensors spread in the network are used in the identification of the target module is completely novel.

The paper is organized as follows. In the next section, we introduce the dynamic network model and we give

the problem statement. In Section 3 we present the identification strategy. In Section 4, we describe the solution scheme based on the ECM algorithm. Additional measurements are added in Section 5, and we present the MCMC based scheme to estimate the target module. Section 6 reports the results of Monte Carlo experiments. Some conclusions end the paper.

1.1 Notation

Given a sequence of scalars $\{a(t)\}_{t=1}^m$, we denote by a its vector representation $a = [a(1) \cdots a(m)]^T \in \Re^m$. Given a vector $a \in \Re^m$, we define by $\mathcal{T}_n(a)$ the $m \times n$ lower triangular Toeplitz matrix whose elements are the entries of a. Lower case letters indicate, in general, column vectors and, when there is no confusion, capital letters indicate their Toeplitz form, so given $a \in \Re^m$, we have that $A = T_n(a)$, where the number n of columns is consistent with the rest of the formula. The symbol "$\otimes$" denotes the standard Kronecker product of two matrices.

2 Problem Statement

2.1 Dynamic networks

We consider dynamic networks that consist of L scalar *internal variables* $w_j(t)$, $j = 1, \ldots, L$ and L scalar external *reference signals* $r_l(t)$, $l = 1, \ldots, L$, that can be manipulated by the user. Some of the reference signal may not be present, i.e., they may be identically zero. Define $\mathcal{R}$ as the set of indices of reference signals that are present. In the dynamic network, the internal variables are considered nodes and transfer functions are the edges. Introducing the vector notation $w(t) := [w_1(t) \ldots w_L(t)]^T$, $r(t) := [r_1(t) \ldots r_L(t)]^T$, the dynamics of the network are defined by the equation

$$w(t) = \mathcal{G}(q)w(t) + r(t), \qquad (1)$$

where

$$\mathcal{G}(q) = \begin{bmatrix} 0 & G_{12}(q) & \cdots & G_{1L}(q) \\ G_{21}(q) & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G_{(L-1)L}(q) \\ G_{L1}(q) & \cdots & G_{L(L-1)}(q) & 0 \end{bmatrix},$$

where $G_{ji}(q)$ is a proper rational transfer function for $j = 1, \ldots, L$, $i = 1, \ldots, L$. The internal variables $w(t)$ are measured with additive white noise, that is

$$\tilde{w}(t) = w(t) + e(t),$$

where $e(t) \in \mathbb{R}^L$ is a stationary zero-mean Gaussian white-noise process with diagonal noise covariance matrix $\Sigma_e = \mathrm{diag}\left\{\sigma_1^2, \ldots, \sigma_L^2\right\}$. We assume that the σ_i^2

are unknown. To ensure stability and causality of the network the following assumptions hold for all networks considered in this paper.

Assumption 2.1 *The network is well posed in the sense that all principal minors of $\lim_{q \to \infty}(I - \mathcal{G}(q))$ are nonzero (Van den Hof et al.; 2013).*

Assumption 2.2 *The sensitivity path $S(q)$*

$$S(q) := (I - \mathcal{G}(q))^{-1}$$

is stable.

Assumption 2.3 *The reference variables $\{r_l(t)\}$ are mutually uncorrelated and uncorrelated with the measurement noise $e(t)$.*

Thus, we can write

$$\tilde{w}(t) = S(q)r(t) + e(t). \qquad (2)$$

We define a $\mathcal{N}_j$ as the set of indices of internal variables that have a direct causal connection to w_j, i.e., $i \in \mathcal{N}_j$ if and only if $G_{ji}(q) \neq 0$. Without loss of generality, we assume that $\mathcal{N}_j = 1, 2, \ldots, p$, where p is the number of direct causal connections to w_j (we may always rename the nodes so that this holds). The goal is to identify module $G_{j1}(q)$ given N measurements of the reference $r(t)$, the "output" $\tilde{w}_j(t)$ and the set of p neighbor signals in $\mathcal{N}_j$. To this end, we express $\tilde{w}_j$, the measured output of module $G_{j1}(q)$ as

$$\tilde{w}_j(t) = \sum_{i \in \mathcal{N}_j} G_{ji}(q)w_i(t) + r_j(t) + e_j(t). \qquad (3)$$

The above equation depends on the internal variables $w_i(t)$, $i \in \mathcal{N}_j$, which we we only have noisy measurement of; these can be expressed as

$$\tilde{w}_i(t) = w_i(t) + e_i(t) = \sum_{l \in \mathcal{R}} S_{il}(q)r_l(t) + e_i(t). \qquad (4)$$

where $S_{il}(q)$ is the transfer function path from reference $r_l(t)$ to output $\tilde{w}_i(t)$. Together, (3) and (4) allow us to express the relevant part of the network, possibly containing feedback loops, as a direct acyclic graph with two blocks connected in cascade. Note that, in general, the first block depends on all other blocks in the network. Therefore, accurate low order parametrization of this block depends on global knowledge of the network.

Example 2.1 *As an example consider the network depicted in Figure 1, where, using (3) and (4), the acyclic graph of Figure 2 can describe the relevant dynamics, when $w_j = w_3$ is the output and we wish to identify $G_{31}(q)$.*
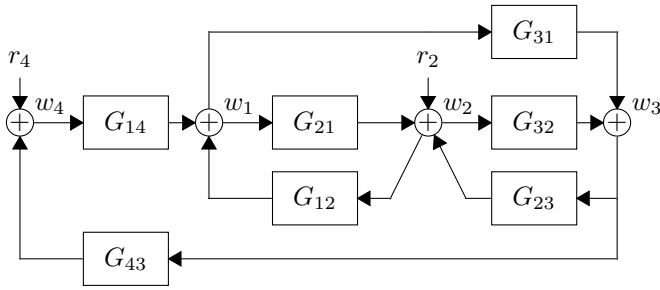
3

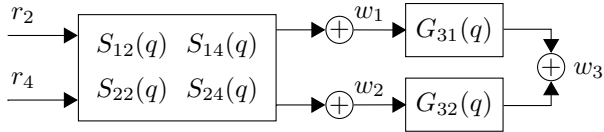Fig. 1. Network example of 4 internal variables and 2 reference signals.

Fig. 2. Direct acyclic graph of part of the network in Figure 1.

In the following, we briefly review two standard methods for closed-loop identification that we will use as a starting point to derive the methodology described in the paper.

2.2 A two stage method

The first stage of the two-stage method (Van den Hof et al.; 2013), proceeds by finding a consistent estimate $\hat{w}_i(t)$ of all nodes $w_i(t)$ in $\mathcal{N}_j$. This is done by high-order modeling of $\{S_{il}\}$ and estimating it from (4) using the prediction error method. The prediction errors are constructed as

$$\varepsilon_i(t, \alpha) = \tilde{w}_i(t) - \sum_{l \in \mathcal{R}} S_{il}(q, \alpha) r_l(t), \qquad (5)$$

where α is a parameter vector. The resulting estimate $S_{il}(q, \hat{\alpha})$ is then used to obtain the node estimate as

$$\hat{w}_i(t) = \sum_{l \in \mathcal{R}} S_{il}(q, \hat{\alpha}) r_l(t). \qquad (6)$$

In a second stage, the module of interest $G_{j1}(q)$ (and the other modules in $\mathcal{N}_j$) is parameterized by θ and estimated from (3), again using the prediction error method. The prediction errors are now constructed as

$$\varepsilon_j(t, \theta) = \tilde{w}_j(t) - r_j(t) - \sum_{i \in \mathcal{N}_j} G_{ji}(q, \theta) \hat{w}_i(t). \quad (7)$$

2.3 Simultaneous minimization of prediction errors

It is useful to briefly introduce the simultaneous minimization of prediction error method (SMPE)
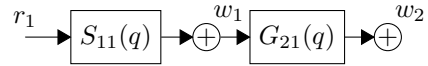

Fig. 3. Basic network of 1 reference signal and 2 internal variables.

(Gunes et al.; 2014). The main idea underlying SMPE is that if, the two prediction errors (5) and (7) are simultaneously minimized, the variance will be decreased (Wahlberg et al.; 2009). In the SMPE method, the prediction error of the measurement $\tilde{w}_j$ depends explicitly on α and is given by

$$\varepsilon_j(t, \theta, \alpha) = \tilde{w}_j(t) - \sum_{i \in \mathcal{N}_j} G_{ji}(q, \theta) \sum_{l \in \mathcal{R}} S_{il}(q, \alpha) r_l(t) . \, (8)$$

The method proceeds to minimize

$$V_N(\theta, \alpha) = \frac{1}{N} \sum_{t=1}^{N} \left[\frac{\varepsilon_j^2(t, \theta, \alpha)}{\sigma_j^2} + \sum_{i \in \mathcal{N}_j} \frac{\varepsilon_i^2(t, \alpha)}{\sigma_i^2} \right] . \quad (9)$$

In (Gunes et al.; 2014), the noise variances are assumed known, and how to estimate the noise variances is not analyzed. As an initial estimate of the parameters θ and α, the minimizers of the two-stage method can be taken.

The main drawback is that the least-squares estimation of S may still induce high variance in the estimates. Additionally, if each of the n_s estimated transfer functions in S is estimated by the first n impulse response coefficients, the number of estimated parameters in S alone is $n_s \cdot n$. Already for relatively small dimensions of S the SMPE method is prohibitively expensive. To handle this, a frequency domain approach is taken in Dankers and Van den Hof (2015). In this paper, we will instead use regularization to reduce the variance and the complexity.

3 Empirical Bayes estimation of the module

In this section we derive our approach to the identification of a specific module based on EB. For ease of exposition, we give a detailed derivation in the one-reference-one-module case. The extension to general dynamic networks follows along similar arguments.

We consider a dynamic network with one non-zero reference signal $r_1(t)$. Without loss of generality, we assume that the module of interest is $G_{21}(q)$, and hence $G_{22}(q), \ldots, G_{2L}(q)$ are assumed zero (We can always rename the signals such that this holds). The setting we consider has been illustrated in Figure 3. We parametrize the target module by means of a parameter vector $\theta \in \mathbb{R}^{n_\theta}$. Using the vector notation introduced in the previous section, we denote by $\tilde{w}_1$ the stacked measurements $\tilde{w}_1(t)$ before the module of interest $G_{21}(q, \theta)$, and by $\tilde{w}_2$

the stacked output of this module $\tilde{w}_2(t)$. We define the impulse response coefficients of $G_{21}(q, \theta)$ by the inverse discrete-time Fourier transform

$$g_\theta(t) := \frac{1}{2\pi} \int_{-\pi}^{\pi} G_{21}(e^{j\omega}, \theta) e^{j\omega t} \, \mathrm{d}\omega. \qquad (10)$$

Similarly we define s_{11} as the impulse response coefficients of $S_{11}(q)$, where $S_{11}(q)$ is, as before, the sensitivity path from from $r_1(t)$ to $w_1(t)$, and $e_1(t)$ and $e_2(t)$ are the measurement noise sources (which we have assumed white and Gaussian). Their variance is denoted by σ_1^2 and σ_2^2, respectively. We rewrite the dynamics as

$$\begin{aligned} \tilde{w}_1 &= S_{11} r_1 + e_1, \\ \tilde{w}_2 &= G_\theta S_{11} r_1 + e_2, \end{aligned} \qquad (11)$$

where G_θ is the $N \times N$ lower triangular Toeplitz matrix of the N first impulse response samples g_θ. The same notation holds for the impulse response s_{11} and its Toeplitz-matrix version $S_{11} = \mathcal{T}_N(s_{11})$. We further rewrite (11) as

$$\begin{aligned} \tilde{w}_1 &= R_1 s_{11} + e_1, \\ \tilde{w}_2 &= G_\theta R_1 s_{11} + e_2. \end{aligned} \qquad (12)$$

where $R_1 = \mathcal{T}_N(r_1)$. For computational purposes, we only consider the first n samples of s_{11}, where n is large enough such that the truncation captures the dynamics of the sensitivity $S_{11}(q)$ well enough. Let $z := [\tilde{w}_1^T \; \tilde{w}_2^T]^T$; we rewrite (12) as

$$z = W_\theta s_{11} + e, \quad W_\theta = \begin{bmatrix} R_1 \\ G_\theta R_1 \end{bmatrix} \quad e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \qquad (13)$$

Note that e is a random vector such that

$$\Sigma_e := \mathrm{E}\left[ee^T\right] = \begin{bmatrix} \sigma_1^2 I & 0 \\ 0 & \sigma_2^2 I \end{bmatrix}. \qquad (14)$$

3.1 Bayesian model of the sensitivity path

To reduce the variance in the sensitivity estimate (and also reduce the number of estimated parameters), we cast our problem in a Bayesian framework and model the sensitivity function as a zero-mean Gaussian stochastic vector (Rasmussen and Williams; 2006), i.e.,

$$p(s_{11}; \lambda, K_\beta) \sim \mathcal{N}(0, \lambda K_\beta). \qquad (15)$$

The structure of the covariance matrix is given by the *first-order stable spline kernel* (Pillonetto and De Nicolao; 2010):

$$\{K_\beta\}_{i,j} = \beta^{\max(i,j)}, \qquad \beta \in [0, 1). \qquad (16)$$

The parameter β regulates the decay velocity of the realizations from (15), whereas, λ tunes their amplitude. In this context, K_β is usually called a *kernel* (due to the connection between Gaussian process regression and the theory of reproducing kernel Hilbert space, see e.g. Rasmussen and Williams (2006) for details) and determines the properties of the realizations of s. In particular, the stable spline kernel enforces smooth and BIBO stable realizations (Pillonetto and De Nicolao; 2010).

3.2 The marginal likelihood estimator

Since s_{11} is assumed stochastic, it admits a probabilistic description jointly with the vector of observations z, parametrized by the vector

$$\eta = \begin{bmatrix} \sigma_1^2 & \sigma_2^2 & \lambda & \beta & \theta \end{bmatrix}. \qquad (17)$$

In particular, having assumed a Gaussian distribution of the noise, the joint description is also Gaussian, that is,

$$p\left(\begin{bmatrix} z \\ s_{11} \end{bmatrix}; \eta\right) \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_z & \Sigma_{zs} \\ \Sigma_{sz} & \lambda K_\beta \end{bmatrix}\right), \qquad (18)$$

where $\Sigma_z = W_\theta \lambda K_\beta W_\theta^T + \Sigma_e$, and $\Sigma_{zs} = \Sigma_{sz}^T = W_\theta \lambda K_\beta$. It is instrumental to derive the posterior distribution of s_{11} given the measurement vector z. It is given by (Anderson and Moore; 1979)

$$p(s_{11}|z; \eta) \sim \mathcal{N}(PW_\theta^T \Sigma_e^{-1} z, P), \qquad (19)$$
$$P = (W_\theta^T \Sigma_e^{-1} W_\theta + (\lambda K_\beta)^{-1})^{-1}, \qquad (20)$$

and it is also parametrized by the vector η.

The module identification strategy we propose in this paper relies on an empirical Bayes approach. We introduce the marginal probability density function (pdf) of the measurements

$$p(z; \eta) = \int p(z, s_{11}) \, ds_{11} \sim \mathcal{N}(0, \Sigma_z), \qquad (21)$$

that is, the pdf of the measurements after having integrating out the dependence on the sensitivity path s_{11}. Then, we can define the (log) marginal likelihood (ML) criterion as the maximum of the marginal pdf defined above

$$\begin{aligned} \hat{\eta} &= \arg\max_\eta p(z; \eta) \\ &= \arg\min_\eta \left(\log\det \Sigma_z + z^T \Sigma_z^{-1} z\right), \end{aligned} \qquad (22)$$

whose solution provides also an estimate of θ and thus of the module of interest.

4 Computation of the solution of the marginal likelihood criterion

Problem (22) is nonlinear and may involve a large number of decision variables, if n_θ is large. In this section, we derive an iterative solution scheme based on the Expectation/Conditional-Maximization (ECM) algorithm (Meng and Rubin; 1993), which is a generalization of the standard Expectation-Maximization (EM) algorithm. In order to employ EM-type algorithms, one has to define a *latent variable*; in our problem, a natural choice is s_{11}. Then, a (local) solution to (22) is achieved by iterating over the following steps:

(E-step) Given an estimate $\hat{\eta}^{(k)}$ (computed at the k-th iteration of the algorithm), compute

$$Q^{(k)}(\eta) := \mathbb{E}\left[\log p(z, s_{11}; \eta)\right], \qquad (23)$$

where the expectation is taken with respect to the posterior of s_{11} when the estimate $\eta^{(k)}$ is used, i.e., $p(s_{11}|z, \hat{\eta}^{(k)})$;

(M-step) Solve the problem

$$\hat{\eta}^{(k+1)} = \arg\max_\eta Q^{(k)}(\eta). \qquad (24)$$

First, we turn our attention on the computation of the E-step, i.e., the derivation of (23). Let $\hat{s}_{11}^{(k)}$ and $\hat{P}^{(k)}$ be the posterior mean and covariance matrix of s_{11}, computed from (19) using $\hat{\eta}^{(k)}$. Define $\hat{S}_{11}^{(k)} := \hat{P}^{(k)} + \hat{s}_{11}^{(k)}\hat{s}_{11}^{(k)T}$. The following proposition provides an expression for the function $Q^{(k)}(\eta)$.

Lemma 4.1 *Let $\hat{\eta}^{(k)} = [\hat{\sigma}_1^{2(k)}\ \hat{\sigma}_2^{2(k)}\ \hat{\lambda}^{(k)}\ \hat{\beta}^{(k)}\ \hat{\theta}^{(k)}]$ be an estimate of η after the k-th iteration of the EM method. Then*

$$Q^{(k)}(\eta) = -\frac{1}{2}Q_0^{(k)}(\sigma_1^2,\sigma_2^2,\theta) - \frac{1}{2}Q_s^{(k)}(\lambda,\beta), \quad (25)$$

where

$$Q_0^{(k)}(\sigma_1^2,\sigma_2^2,\theta) = \Big(\log\det\Sigma_e + z^T\Sigma_e^{-1}z - 2z^T W_\theta\hat{s}_{11}^{(k)}$$
$$+ Tr\Big\{W_\theta^T\Sigma_e^{-1}W_\theta\hat{S}_{11}^{(k)}\Big\}\Big), \qquad (26)$$

$$Q_s^{(k)}(\lambda,\beta) = \log\det\lambda K_\beta + Tr\Big\{(\lambda K_\beta)^{-1}\hat{S}_{11}^{(k)}\Big\}. \qquad (27)$$

Having computed the function $Q^{(k)}(\eta)$, we now focus on its maximization. We first note that the decomposition (25) shows that the kernel hyperparameters can be updated independently of the rest of the parameters:

Proposition 4.1 *Define*

$$Q_\beta(\beta) = \log\det K_\beta + n\log Tr\Big\{K_\beta^{-1}\hat{S}_{11}^{(k)}\Big\}. \qquad (28)$$

Then

$$\hat{\beta}^{(k+1)} = \arg\min_{\beta\in[0,1)} Q_\beta(\beta), \qquad (29)$$

$$\hat{\lambda}^{(k+1)} = \frac{1}{n}Tr\Big\{K_{\hat{\beta}^{(k+1)}}^{-1}\hat{S}_{11}^{(k)}\Big\}. \qquad (30)$$

Therefore, the update of the scaling hyperparameter is available in closed-form, while the update of β requires the solution of a scalar optimization problem in the domain $[0, 1]$, an operation that requires little computational effort, see Bottegal et al. (2016) for details.

We are left with the maximization of the function $Q_0^{(k)}(\sigma_1^2,\sigma_2^2,\theta)$. In order to simplify this step, we split the optimization problem into constrained subproblems that involve fewer decision variables. This operation is justified by the ECM paradigm, which, under mild conditions (Meng and Rubin; 1993), guarantees the same convergence properties of the EM algorithm even when the optimization of $Q^{(k)}(\eta)$ is split into a series of constrained subproblems. In our case, we decouple the update of the noise variances from the update of θ. By means of the ECM paradigm, we split the maximization of $Q_0^{(k)}(\sigma_1^2,\sigma_2^2,\theta)$ in a sequence of two constrained optimization subproblems:

$$\hat{\theta}^{(k+1)} = \arg\max_\theta Q_0^{(k)}(\sigma_1^2,\sigma_2^2,\theta) \qquad (31)$$
$$\text{s.t. } \sigma_1^2 = \hat{\sigma}_1^{2(k)},\ \sigma_2^2 = \hat{\sigma}_2^{2(k)},$$
$$\hat{\sigma}_1^{2(k+1)},\ \hat{\sigma}_2^{2(k+1)} = \arg\max_{\sigma_1^2,\sigma_2^2} Q_0^{(k)}(\sigma_1^2,\sigma_1^2,\theta) \qquad (32)$$
$$\text{s.t. } \theta = \hat{\theta}^{(k+1)}.$$

The following result provides the solution of the above problems.

Proposition 4.2 *Introduce the matrix $D \in \mathbb{R}^{N^2\times N}$ such that $Da = \mathrm{vec}(\mathcal{T}_N(a))$, for any $a\in\mathbb{R}^N$. Define*

$$\hat{A}^{(k)} = D^T(R_1\hat{S}_{11}^{(k)}R_1^T\otimes I_N)D \qquad (33)$$
$$\hat{b}^{(k)} = \mathcal{T}_N(R_1\hat{s}_{11}^{(k)})^T\tilde{w}_2. \qquad (34)$$

Then

$$\hat{\theta}^{(k+1)} = \arg\min_\theta g_\theta^T\hat{A}^{(k)}g_\theta - 2\hat{b}^{(k)T}g_\theta. \qquad (35)$$

The closed form updates of the noise variances are as

follows

$$\hat{\sigma}_1^{2(k+1)} = \frac{1}{N}\Big(\|\tilde{w}_1 - R_1\hat{s}_{11}^{(k)}\|_2^2 + Tr\Big\{R_1\hat{P}^{(k)}R_1^T\Big\}\Big)\,,$$
$$\hat{\sigma}_2^{2(k+1)} = \frac{1}{N}\Big(\|\tilde{w}_2 - G_{\hat{\theta}^{(k+1)}}R_1\hat{s}_{11}^{(k)}\|_2^2$$
$$+ Tr\Big\{G_{\hat{\theta}^{(k+1)}}R_1\hat{P}^{(k)}R_1^T G_{\hat{\theta}^{(k+1)}}^T\Big\}\Big)\,. \quad (36)$$

Each variance is the result of the sum of one term that measures the adherence of the identified systems to the data and one term that compensates for the bias in the estimates introduced by the Bayesian approach. The update of the parameter θ involves a (generally) nonlinear least-squares problem, which can be solved using gradient descent strategies. Note that, in case the impulse response g_θ is linearly parametrized (e.g., it is an FIR system or orthonormal basis functions are used (Wahlberg; 1991)), then the update of θ is also available in closed-form.

Example 4.1 *Assume that the linear parametrization $g_\theta = L\theta$, $L \in \mathbb{R}^{N \times n_\theta}$, is used, then*

$$\hat{\theta}^{(k+1)} = \Big(L^T\hat{A}^{(k)}L\Big)^{-1}L^T\hat{b}^{(k)}\,. \quad (37)$$

4.1 Identification algorithm

The proposed method for module identification can be summarized in the following steps.

(1) Find an initial estimate of $\hat{\eta}^{(0)}$, set $k = 0$.
(2) Compute $\hat{s}_{11}^{(k)}$ and $\hat{P}^{(k)}$ from (19).
(3) Update the kernel hyperparameters using (30), (29).
(4) Update the vector θ solving (35).
(5) Update the noise variances from (36).
(6) Check if the algorithm has converged. If not, set $k = k + 1$ and go back to step 2.

The method can be initialized in several ways. One option is to first estimate $\hat{S}_{11}(q)$ by an empirical Bayes method using only r_1 and $\tilde{w}_1$. Then, $\hat{w}_1$ is constructed from (6), using the obtained $\hat{S}_{11}(q)$. Finally, G is estimated using the prediction error method, using $\hat{w}_1$ as input and $\tilde{w}_2$ as output.

4.2 Extension to general structures

In this section, we generalize the previous algorithm to a general network structure with $m \leq L$ reference signals $\{r_{l_1}(t), \ldots, r_{l_m}(t)\}$, and $p \leq L$ modules $\{G_{j1}(q), \ldots, G_{jp}(q)\}$ sharing the same output $\tilde{w}_j(t)$ as

the module of interest, and modeled in time domain as $g_{\theta_1}, \ldots, g_{\theta_p}$. For any $i = 1, \ldots, p$, we can write

$$\tilde{w}_i = R_{l_1}s_{il_1} + \ldots + R_{l_m}s_{il_m} + e_{k_i} = \mathbf{R}s_i + e_{k_i}\,, \quad (38)$$

where $\mathbf{R} := [R_{l_1} \ldots R_{l_m}]$ and $s_i = [s_{il_1}^T \ldots s_{il_m}^T]^T$. Using these definitions we can also write (cf. (3))

$$\tilde{w}_j = r_j + G_{\theta_1}\mathbf{R}s_1 + \ldots + G_{\theta_p}\mathbf{R}s_p + e_j\,. \quad (39)$$

Defining also $\mathbf{w} = [\tilde{w}_1^T \ldots \tilde{w}_p^T]^T$, $\mathbf{s} = [s_1^T \ldots s_p^T]^T$, $G_\theta = [G_{\theta_1} \ldots G_{\theta_p}]$, $\mathbf{e_w} = [e_1^T \ldots e_p^T]^T$, we obtain the following expression for the network dynamics

$$\mathbf{w} = (I_p \otimes \mathbf{R})\mathbf{s} + \mathbf{e_w}$$
$$\tilde{w}_j - r_j = G_\theta(I_p \otimes \mathbf{R})\mathbf{s} + e_j\,, \quad (40)$$

or, with $\mathbf{z} = [\mathbf{w}^T\ (\tilde{w}_j - r_j)^T]^T$

$$\mathbf{z} = \mathbf{W}_\theta\mathbf{s} + \mathbf{e},\ \mathbf{W}_\theta = \begin{bmatrix} (I_p \otimes \mathbf{R}) \\ G_\theta(I_p \otimes \mathbf{R}) \end{bmatrix},\ \mathbf{e} = \begin{bmatrix} \mathbf{e_w} \\ e_j \end{bmatrix}\,. \quad (41)$$

Each sensitivity path s_{il} is given a prior of the form (15), with hyperparameters λ_{il} and β_{il}, assuming mutual independence between the sensitivity paths. Although it may appear more sensible to incorporate some correlation among the sensitivity paths, at present, it is not clear how this can be done using Gaussian priors. Some recent work suggests to enrich the stable spline kernel with a component enforcing low McMillan degree (Prando et al.; 2014). Furthermore, as we will see, assuming independent priors allows the kernel hyperparameters to be updated independently. Introducing Λ as the diagonal matrix with elements corresponding to $\{\lambda_{il}\}$, and similarly, defining $\mathbf{K}_\beta$ with diagonal elements $\{K_{\beta_{il}}\}$, we have

$$p(\mathbf{s};\ \Lambda,\ \mathbf{K}_\beta) \sim \mathcal{N}\left(0,\ (\Lambda \otimes I_n)\mathbf{K}_\beta\right)\,. \quad (42)$$

We collect all the parameters characterizing the model into the vector η. It follows that

$$p(\mathbf{z};\ \eta) \sim \mathcal{N}(0,\ \Sigma_\mathbf{z})\,, \quad (43)$$

where $\Sigma_\mathbf{z} = \mathbf{W}_\theta(\Lambda \otimes I_n)\mathbf{K}_\beta\mathbf{W}_\theta^T + \Sigma_\mathbf{e}$, and

$$\Sigma_\mathbf{e} = \text{diag}\left\{\sigma_1^2,\ \ldots,\ \sigma_p^2,\ \sigma_j^2\right\} \otimes I_N\,. \quad (44)$$

Therefore, we can define the following ML criterion

$$\hat{\eta} = \arg\max_\eta \log p(\mathbf{z};\ \eta)\,. \quad (45)$$

Having set the notation, we outline the ECM algorithm for this general setting below. To this end, note that

$$p(\mathbf{s}|\mathbf{z};\ \eta) = \mathcal{N}(\hat{\mathbf{s}},\ \mathbf{P})\,, \quad (46)$$

where

$$\hat{\mathbf{s}} = \mathbf{P}\mathbf{W}_\theta^T \Sigma_{\mathbf{e}}^{-1}\mathbf{z}\,, \qquad (47)$$

$$\mathbf{P} = \left(\mathbf{W}_\theta^T \Sigma_{\mathbf{e}}^{-1}\mathbf{W}_\theta + ((\Lambda \otimes I_n)\mathbf{K}_\beta)^{-1}\right)^{-1}\,. \qquad (48)$$

We use again the notation $\hat{\mathbf{s}}^{(k)}$, $\hat{\mathbf{P}}^{(k)}$, $\hat{\mathbf{S}}^{(k)}$ to mean the estimates of the corresponding quantities at iteration k.

Proposition 4.3 *Let η collect all the parameters characterizing (45), and let $\eta^{(k)}$ be its estimate after the k-th iteration of the ECM method. Then the estimate $\eta^{(k+1)}$ is obtained by means of the following updates.*

Hyperparameters: *Define*

$$Q_{\beta_{ij}}(\beta) = \log\det K_\beta + n\log Tr\left\{K_\beta^{-1}\hat{S}_{ij}^{(k)}\right\}\,, \ (49)$$

where $\hat{S}_{ij}^{(k)}$ is the $n\times n$ diagonal block of $\hat{S}^{(k)}$ corresponding to the path s_{ij}. Then λ_{ij} and β_{ij} are updated as in Proposition 4.1, for any i, j.

Module parameters: *Define*

$$\hat{\mathbf{A}}^{(k)} := (I_p \otimes D)^T \left(\bar{\mathbf{R}}\hat{\mathbf{S}}^{(k)}\bar{\mathbf{R}}^T \otimes I_N\right)(I_p \otimes D)\,, \ (50)$$

$$\hat{\mathbf{b}}^{(k)T} := (\tilde{w}_j - r_j)^T \left[\mathcal{T}_N(\mathbf{R}\hat{s}_1^{(k)}) \dots \mathcal{T}_N(\mathbf{R}\hat{s}_p^{(k)})\right]\,, (51)$$

where D is as in Proposition 4.2 and $\bar{\mathbf{R}} = I_p \otimes \mathbf{R}$. Then

$$\hat{\theta}^{(k+1)} = \arg\min_\theta \mathbf{g}_\theta^T \hat{\mathbf{A}}^{(k)}\mathbf{g}_\theta - 2\hat{\mathbf{b}}^{(k)T}\mathbf{g}_\theta\,, \qquad (52)$$

where $\mathbf{g}_\theta := [g_{\theta_1}^T \ \dots \ g_{\theta_p}^T]^T$.

Noise variances:

$$\hat{\sigma}_i^{2(k+1)} = \frac{1}{N}\left(\|\tilde{w}_i - \mathbf{R}\hat{s}_i^{(k)}\|_2^2 + Tr\left\{\mathbf{R}\hat{\mathbf{P}}_i^{(k)}\mathbf{R}^T\right\}\right)$$

$$\hat{\sigma}_j^{2(k+1)} = \frac{1}{N}\left(\|\tilde{w}_j - r_j - G_{\hat{\theta}^{(k+1)}}(I \otimes \mathbf{R})\hat{s}^{(k)}\|_2^2 \right.$$
$$\left. + Tr\left\{G_{\hat{\theta}^{(k+1)}}\bar{\mathbf{R}}\hat{\mathbf{P}}^{(k)}\bar{\mathbf{R}}^T G_{\hat{\theta}^{(k+1)}}^T\right\}\right)\,, \qquad (53)$$

where $\mathbf{P}_i^{(k)}$ is the $nm \times nm$ diagonal block of $\mathbf{P}^{(k)}$, corresponding to the covariance matrix of $\hat{s}_i^{(k)}$.

5 Including additional sensors

By using the kernel-based approach adopted above, the sensitivity paths could be modeled with only a few hyperparameters while still keeping the module of interest parametric. One potential benefit with this approach is that including another reference signal will not increase the number of estimated parameters significantly. Although the complexity of the problem increases slightly,
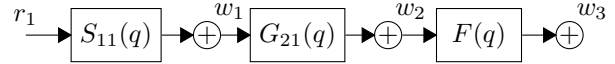

Fig. 4. Basic network of 1 reference signal and 3 internal variables.

only a few extra hyperparameters need to be estimated and the dimensions of (35) remain the same in the update of θ.

As reference signals can be added with little effort, a natural question is if also output measurements "downstream" of the module of interest can be added with little effort. In Example 2.1 the measurement w_4 is such a measurement that, with the same strategy as before, can be expressed as

$$w_4(t) = G_{43}(q)w_3(t) + r_4(t)\,. \qquad (54)$$

Using this measurement for the purpose of identification would require the identification of $G_{43}(q)$ in addition to the previously considered modules. The signal $w_4(t)$ contains information about $w_3(t)$, and thus information about the module of interest. The price we have to pay for this information is the additional parameters to estimate and, as we will see, another layer of complexity.

To extend the previous framework to include additional measurements after the module of interest, let us consider the case where we would like to include only one additional measurement, in this context denoted by $\tilde{w}_3(t)$; the generalization to more sensors is straightforward but notationally heavy. Let the path linking the target module to the additional sensor be denoted by $F(q)$, with impulse response f. Furthermore, let us for simplicity consider the one-reference-signal-one-input case again, i.e., (11), (12). The setting we consider has been illustrated in Figure 4. We model also this module using a Bayesian framework by interpreting f as a zero-mean Gaussian stochastic vector, i.e.,

$$p(f;\lambda_f, K_{\beta_f}) \sim \mathcal{N}(0, \lambda_f K_{\beta_f})\,, \qquad (55)$$

where again K_{β_f} is the first-order stable spline kernel (16). We introduce the following variables

$$\sigma = \begin{bmatrix} \sigma_1^2 & \sigma_2^2 & \sigma_3^2 \end{bmatrix}\,, \qquad (56)$$

$$z = \begin{bmatrix} \tilde{w}_1^T & \tilde{w}_2^T & \tilde{w}_3^T \end{bmatrix}^T\,, \qquad (57)$$

$$z_f = \tilde{w}_3\,. \qquad (58)$$

For given vales of θ, s and f, we construct

$$W_s = \begin{bmatrix} R \\ G_\theta R \\ F G_\theta R \end{bmatrix}, \qquad (59)$$

$$W_f = \mathcal{T}_N(G_\theta R s_{11}), \qquad (60)$$

$$\Sigma = \operatorname{diag}\{\sigma\} \otimes I_N. \qquad (61)$$

Notice that the last internal variable w_3 can be expressed as

$$\begin{aligned}
w_3 &= F G_\theta S_{11} r \\
&= G_\theta F S_{11} r \\
&= G_\theta F R s_{11} \\
&= G_\theta R F s_{11} \\
&= G_\theta R v, \qquad (62)
\end{aligned}$$

where commutation of the matrices follows from the fact that they are lower-triangular Toeplitz matrices, and $v := F s_{11}$. For ease of exposition, we will also use the notation $v = f * s_{11}$.

The key difficulty in this setup is that the description of the measurements and the system description with both s_{11} and f no longer admit a jointly Gaussian probabilistic model, because v in (62) is the result of the convolution of two Gaussian vectors. In fact, a closed-form expression is not available. This fact has a detrimental effect in our empirical Bayes approach, because the marginal likelihood estimator of

$$\eta = \begin{bmatrix} \sigma & \lambda_s & \beta_s & \lambda_f & \beta_f & \theta \end{bmatrix},$$

where λ_s, β_s are the hyperparameters of the prior of s_{11}, that is

$$\hat{\eta} = \arg\max_\eta p(z; \eta) \qquad (63)$$

$$= \arg\max_\eta \int p(z, s_{11}, f; \eta)\, \mathrm{d}s_{11}\, \mathrm{d}f, \qquad (64)$$

does not admit an analytical expression, since the integral (64) is intractable. To treat this problem, again we resort to EM-type methods. In this case, the latent variables to add to the problem are both s_{11} and f, so that the EM method has to alternate between the following steps.

(E-step) Given an estimate $\hat{\eta}^{(k)}$ (computed at the k-th iteration of the algorithm), compute

$$Q^{(k)}(\eta) := \mathbb{E}\left[\log p(z, s_{11}, f; \eta)\right], \qquad (65)$$

where the expectation is taken with respect to the target distribution when the estimate $\eta^{(k)}$ is used, i.e., $p(s_{11}, f|z, \hat{\eta}^{(k)})$;

(M-step) Solve the problem

$$\hat{\eta}^{(k+1)} = \arg\max_\eta Q^{(k)}(\eta). \qquad (66)$$

While the M-Step remains substantially unchanged, the E-step requires more attention. Now, it requires the computation of the integral

$$\mathbb{E}\left[\log p(z, s_{11}, f; \eta)\right] = \qquad (67)$$
$$\int \log p(z, s_{11}, f; \eta) p(s_{11}, f|z, \hat{\eta}^{(k)})\, \mathrm{d}s_{11}\, \mathrm{d}f,$$

which does not admit an analytical solution, because the posterior distribution $p(s_{11}, f|z, \hat{\eta}^{(k)})$ is non-Gaussian (it does not have an analytical form, in fact). However, using Markov Chain Monte Carlo (MCMC) techniques, we can compute an approximation of the integral by sampling from the joint posterior density (also called a target distribution)

$$p(s_{11}, f|z; \eta). \qquad (68)$$

As pointed out before, (68) does not admit a closed-form expression and hence direct sampling is a hard task. However, if it is easy to draw samples from the conditional probability distributions, samples of (68) can be easily drawn using the Gibbs sampler. In Gibbs sampling, each conditional is considered the state of a Markov chain; by iteratively drawing samples from the conditionals, the Markov chain will converge to its stationary distribution, which corresponds to the target distribution. In our problem, the conditionals of (68) are as follows

- $p(s_{11}|f, z; \eta)$. Using (59), we write the linear model

$$z = W_s s_{11} + e, \qquad (69)$$

where $e = [e_1^T \ e_2^T \ e_3^T]^T$. Then, given f, the vectors s_{11} and z are jointly Gaussian, so that

$$p(s|f, z; \eta) \sim \mathcal{N}(m_s, P_s), \qquad (70)$$

with

$$P_s = \left(W_s^T \Sigma^{-1} W_s + (\lambda_s K_{\beta_s})^{-1}\right)^{-1}$$
$$m_s = P_s W_s^T \Sigma^{-1} z.$$

- $p(f|s, z; \eta)$. Given s and r, all sensors but the last becomes redundant. Using (60) we write the linear model

$$z_f = W_f f + e_3, \qquad (71)$$

which shows that

$$p(f|s_{11}, z; \eta) \sim \mathcal{N}(m_f, P_f), \qquad (72)$$

with

$$P_f = \left(\frac{W_f^T W_f}{\sigma_3^2} + (\lambda_f K_{\beta_f})^{-1} \right)^{-1}$$

$$m_f = P_f \frac{W_f^T}{\sigma_3^2} z_f \,.$$

The following algorithm summarizes the Gibbs sampler used for dynamic network identification.

Algorithm 1 *Gibbs sampler for a dynamic network. Initialization: compute initial value of s^0 and f^0. For $k = 1$ to $M + M_0$:*

(1) Draw the sample s^k from $p(s|f^{k-1}, z; \eta)$;
(2) Draw the sample f^k from $p(f|s^k, z; \eta)$;

In this algorithm, M_0 is the number of initial samples that are discarded, which is also known as the *burn-in* (Meyn and Tweedie; 2009). These samples are discarded since the Markov chain needs a certain number of samples to converge to its stationary distribution.

5.1 The ECM method with additional sensor

We now discuss the computation of the E-step and the CM-steps using the Gibbs sampler scheme introduce above.

Proposition 5.1 *Introduce the mean and covariance quantities*

$$s_s^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} s^k, \qquad (73)$$

$$f_s^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} f^k, \qquad (74)$$

$$v_s^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} v^k, \qquad (75)$$

$$P_s^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} (s^k - s_s^M)(s^k - s_s^M)^T, \qquad (76)$$

$$P_f^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} (f^k - f_s^M)(f^k - f_s^M)^T, \qquad (77)$$

$$P_v^M = \frac{1}{M} \sum_{k=M_0+1}^{M_0+M} (v^k - v_s^M)(v^k - v_s^M)^T, \qquad (78)$$

where s^k, f^k and $v^k = s^k * f^k$ are samples drawn using Algorithm 1.

Define

$$\tilde{Q}_s(\lambda, \beta, x, X) := \log \det \lambda K_\beta \\ + Tr\{(\lambda K_\beta)^{-1}(xx^T + X)\} \,,$$

$$\tilde{Q}_z(\sigma^2, z, x, X) := N \log \sigma^2 + \frac{1}{\sigma^2} \|z - Rx\|_2^2 \\ + \frac{1}{\sigma^2} Tr\{RXR^T\} \,,$$

$$\tilde{Q}_f(\sigma^2, z, \theta, x, X) := N \log \sigma^2 + \frac{1}{\sigma^2} \|z - G_\theta Rx\|_2^2 \\ + \frac{1}{\sigma^2} Tr\{G_\theta RXR^T G_\theta^T\} \,.$$

Then

$$-2Q^{(k)}(\eta) = \lim_{M \to \infty} \tilde{Q}_s(\lambda_s, \beta_s, s_s^M, P_s^M), \\ + \tilde{Q}_s(\lambda_f, \beta_f, f_s^M, P_f^M), \\ + \tilde{Q}_z(\sigma_1^2, \tilde{w}_1, s_s^M, P_s^M), \\ + \tilde{Q}_f(\sigma_2^2, \tilde{w}_2, \theta, s_s^M, P_s^M), \\ + \tilde{Q}_f(\sigma_3^2, \tilde{w}_3, \theta, v_s^M, P_v^M). \qquad (79)$$

The CM-steps are now very similar to the previous method and follows by similar reasoning as in the proof of Proposition 4.2.

Proposition 5.2 *Let $\hat{\eta}^{(k)}$ be the parameter estimate obtained at the k:th iteration. Define $S_s^M = s_s^M (s_s^M)^T + P_s^M$, $S_v^M = v_s^M (v_s^M)^T + P_v^M$,*

$$\hat{A}_s = D^T(RS_s^M R^T \otimes I_N)D \,, \\ \hat{A}_v = D^T(RS_v^M R^T \otimes I_N)D \,, \\ \hat{b}_s = \mathcal{T}_N(Rs_s^M)^T \tilde{w}_2 \,, \\ \hat{b}_v = \mathcal{T}_N(Rv_s^M)^T \tilde{w}_3 \,.$$

Then the updated parameter vector $\hat{\eta}^{(k+1)}$ is obtained as follows

$$\hat{\theta}^{(k+1)} = \arg\min_\theta g_\theta^T \left(\frac{1}{\sigma_2^2} \hat{A}_s + \frac{1}{\sigma_3^2} \hat{A}_v \right) g_\theta \\ - 2 \left(\frac{1}{\sigma_2^2} \hat{b}_s^T + \frac{1}{\sigma_3^2} \hat{b}_v^T \right) g_\theta \,. \qquad (80)$$

The closed form updates of the noise variances are

$$\hat{\sigma}_1^{2(k+1)} = \frac{1}{N} \left(\|\tilde{w}_1 - Rs_s^M\|_2^2 + Tr\{RP_s^M R^T\} \right) \,,$$

$$\hat{\sigma}_2^{2(k+1)} = \frac{1}{N} \left(\|\tilde{w}_2 - G_{\hat{\theta}^{(k+1)}} Rs_s^M\|_2^2 \\ + Tr\{G_{\hat{\theta}^{(k+1)}} RP_s^M R^T G_{\hat{\theta}^{(k+1)}}^T\} \right) \,,$$

$$\hat{\sigma}_3^{2(k+1)} = \frac{1}{N} \left(\|\tilde{w}_3 - G_{\hat{\theta}^{(k+1)}} Rv_s^M\|_2^2 \\ + Tr\{G_{\hat{\theta}^{(k+1)}} RP_v^M R^T G_{\hat{\theta}^{(k+1)}}^T\} \right) \,. \qquad (81)$$

The kernel hyperparameters are updated through (29) *and* (30) *for both* s_{11} *and* f.

5.2 Identification algorithm

The proposed method for module identification can be summarized in the following steps.

(1) Find an initial estimate of $\hat{\eta}^{(0)}$, set $k = 0$.
(2) Compute the quantities (73)-(78) using Algorithm 1.
(3) Update the kernel hyperparameters using (30), (29).
(4) Update the vector θ solving (80).
(5) Update the noise variances from (81).
(6) Check if the algorithm has converged. If not, set $k = k + 1$ and go back to step 2.

As can be seen, the main difference with the one-input-one-sensor algorithm (see Section 5.2) is that Step 2 of the algorithm requires a heavier computational burden, because of the integration via Gibbs sampling. Nevertheless, as will be seen in the next section, this pays off in terms of performance in identifying the target module.

6 Numerical experiments

In this section, we present results from two Monte Carlo simulations to illustrate the performance of the proposed method, which we abbreviate as *Network Empirical Bayes* (NEB) and its extension NEBX outlined in Section 5, and we compare with SMPE (see Section 2.3). We consider the network case of Example 2.1 and a simple closed loop network. The reference signals used are zero-mean unit-variance Gaussian white noise. The noise signals e_k are zero-mean Gaussian white noise with variances such that noise to signal ratios $\mathrm{Var}\, w_k / \mathrm{Var}\, e_k$ are constant. The setting of the compared methods are provided in some more details below, where the model order of the plant $G(q)$ is known for both the SMPE method and the proposed NEB method.

NEB: The method is initialized by the two-stage method. First, $\hat{S}(q)$ is estimated by least-squares. Second, G is estimated using MORSM (Everitt, Galrinho and Hjalmarsson; 2016) from the simulated signal $\hat{w}$ obtained from (6) and $\tilde{w}_j$. MORSM is an iterative method that is asymptotically efficient for open loop data. Then, the iterative method outlined in Section 4.1 is employed with the stopping criteria $\left\| \hat{\eta}^{(k+1)} - \hat{\eta}^{(k)} \right\| / \left\| \hat{\eta}^{(k)} \right\| < 10^{-10}$.

NEBX: The method is initialized by NEB. f^0 is obtained by an empirical Bayes method using simulated input and measured output of f. Then, the iterative method outlined in Section 5 is employed with the stopping criteria $\left\| \hat{\eta}^{(k+1)} - \hat{\eta}^{(k)} \right\| / \left\| \hat{\eta}^{(k)} \right\| < 10^{-10}$, or a maximum of 50 iterations.

SMPE: The method is initialized by the two-stage method, exactly as NEB. Then, the cost function (9), with a slight modification, is minimized. The modification of the cost function comes from that, as mentioned before, the SMPE method assumes that the noise variances are known. To make the comparison fair, also the noise variances need to be estimated. By maximum likelihood arguments, the logarithm of the determinant of the complete noise covariance matrix is added to the cost function (9) and the noise variances are included in θ, the vector of parameters to estimate. The tolerance is set to $\left\| \hat{\theta}^{(k+1)} - \hat{\theta}^{(k)} \right\| / \left\| \hat{\theta}^{(k)} \right\| < 10^{-10}$.

The simulations were run in Julia, a high-level, high-performance dynamic programming language for technical computing (Bezanson et al.; 2017).

6.1 Closed-loop identification

The first Monte Carlo simulation is from a system operating in closed loop with an unknown low order controller with $N = 200$ data samples. This setting is slightly different to the standard closed-loop setting in that the measurement noise of $\tilde{w}_2$ is not fed back in the loop, and that the signals w_1 and w_2 are treated completely symmetric. The noise to signal ratio are all set to 1. The true plant and true controller are chosen such that the sensitivity function $S(q^{-1})$ has an impulse response that can be well approximated by $n = 100$ impulse response coefficients. The closed loop is depicted in Figure 5, where
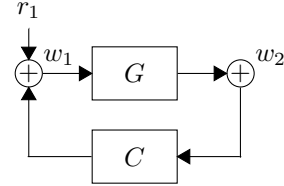

Fig. 5. Closed loop network of first Monte Carlo simulation.

$$G(q, \theta) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}, \qquad (82)$$

The controller C is given by

$$C(q, \theta) = \frac{0.8 + 0.4 q^{-1} - 0.5 q^{-2}}{1 + 0.5 q^{-1} + 0.2 q^{-2}}, \qquad (83)$$

with the parameter vector $\theta = [b_1, b_2, a_1, a_2]$, and true parameters $\theta^0 = [0.4, 0.5, -0.4, 0.3]$.

The two methods are compared using the fit of the impulse response coefficients of g according to

$$FIT = 1 - \frac{\left\| g^0 - \hat{g} \right\|_2}{\left\| g^0 \right\|_2} \qquad (84)$$

11

For this example, the proposed NEB method achieves a higher fit, on average, than the SMPE method, cf. the box plot of Figure 6. Comparing the fits obtained at each Monte Carlo run (see Figure 7), it can be seen that NEB consistently performs at least as good as SMPE for almost every Monte Carlo run and in some runs considerably better. From the sample means and variance reported in Table 1, it can be seen that, in general, the estimates produced by NEB have smaller variance than SMPE while their mean values are similar.
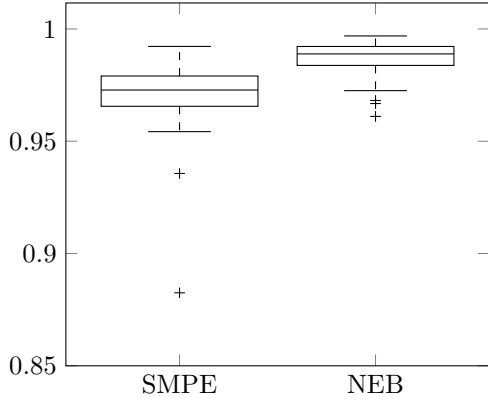

Fig. 6. Box plot of the fit of the impulse response of G obtained by the SMPE, and NEB methods respectively.
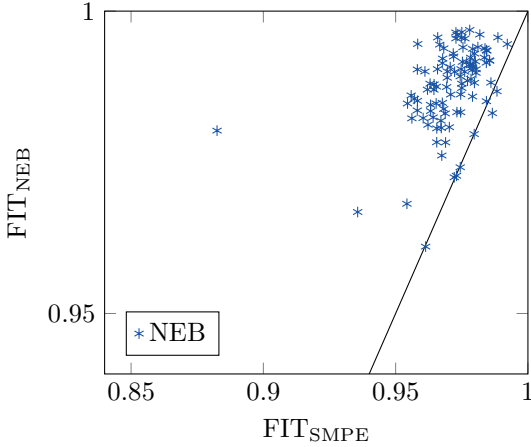

Fig. 7. Each fit of the impulse response coefficients of G for NEB compared with SMPE for 100 Monte Carlo simulations. The black line represents $y = x$, i.e., when SMPE performs equally good as NEB. Note the scaling of the x-axis of this figure.

6.2 Dynamic network example

This Monte Carlo simulation compares the NEB method and NEBX with the SMPE method on data from the network of Example 2.1, illustrated in Figure 1, where each of the modules are of second order, i.e.,

$$G_{ij}(q) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}},$$

for a set of parameters that were chosen such that all modules are stable and $\{S_{12}(q), S_{24}(q), S_{22}(q), S_{24}(q)\}$ are stable and can be well approximated with 70 impulse response coefficients. Two reference signals, $r_2(t)$ and $r_4(t)$ are available and $N = 200$ data samples are used with the goal to estimate $G_{31}(q)$ and G_{32}. In total 6 transfer functions are estimated, $\{S_{12}(q), S_{24}(q), S_{22}(q), S_{24}(q), G_{31}(q)$ and $G_{32}(q)\}$, where $\{S_{12}(q), S_{24}(q), S_{22}(q), S_{24}(q)\}$ are each parameterized by $n = 75$ impulse response coefficients in all methods. For NEBX also $G_{43}(q)$ is estimated by $n = 75$ impulse response coefficients. The noise to signal ratio at each measurement is set to $\text{Var } w_k/\text{Var } e_k = 0.1$ and the additional measurement used in NEBX has a lower noise to signal ratio of $\text{Var } w_4/\text{Var } e_4 = 0.01$.

The fits of the impulse responses of G_{31} and G_{32} for the experiment are shown as a boxplot in Figure 8 and Figure 10 respectively. Comparing the fits obtained at each Monte Carlo run (see Figure 11 and Figure 11), the proposed NEB and NEBX methods are competitive with the SMPE method for this network. In many cases, the SMPE method failed to produce a reasonable estimate as 10 percent of the Monte Carlo runs gave a negative fit and were removed before the impulse response fits, boxplots and parameter sample means and variances were computed. From the sample means and variance reported in Table 2 and Table 3, it can be seen that, in general, the estimates produced by NEB and NEBX have, in general, significantly smaller variance than SMPE, while the mean values are roughly the same. Recalling that one of the motivations of the proposed methods was to reduced the variance induced by the high order modeling of the sensitivity paths, both the closed-loop example and network example gives some support for this motivation.

In almost all of the Monte Carlo runs, NEBX outperformed NEB in this simulation. However, NEBX is significantly more computationally expensive than NEB.
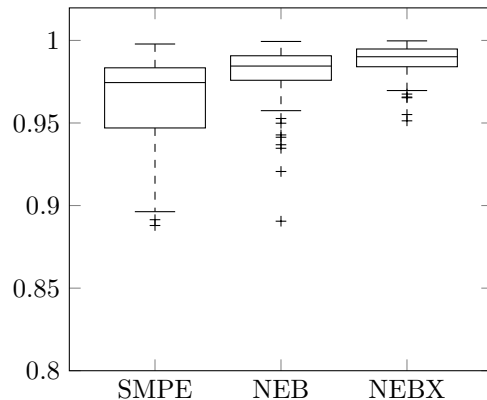

Fig. 8. Box plot of the fit of the impulse response of G_{31} obtained by the methods SMPE, NEB and NEBX respectively.

Table 1
Sample mean and sample variance of the parameters estimates for $\hat{G}$ for compared methods.

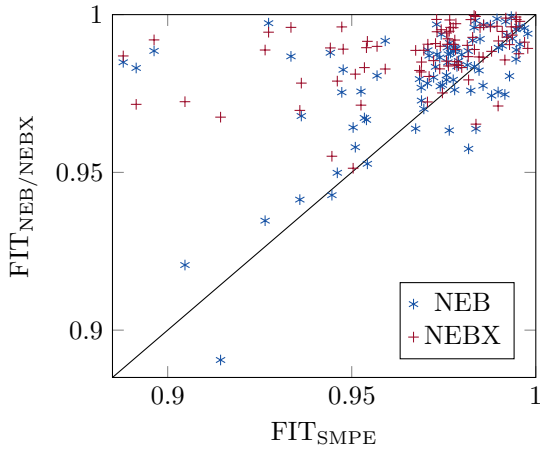| Method	$b_1^0 = 0.2$		$b_2^0 = 0.3$		$a_1^0 = 0.4$		$a_2^0 = 0.5$	
	$E_s\,\hat{b}_1$	$N \cdot \text{Var}_s\,\hat{b}_1$	$E_s\,\hat{b}_2$	$N \cdot \text{Var}_s\,\hat{b}_2$	$E_s\,\hat{a}_1$	$N \cdot \text{Var}_s\,\hat{a}_1$	$E_s\,\hat{a}_2$	$N \cdot \text{Var}_s\,\hat{a}_2$
SMPE	0.21	0.43	0.31	0.93	0.50	3.4	0.16	2.8
NEB	0.20	0.22	0.31	0.26	0.68	2.9	0.23	2.0

Table 2
Sample mean and sample variance of the parameters estimates for $\hat{G}_{31}$ for the three compared methods.

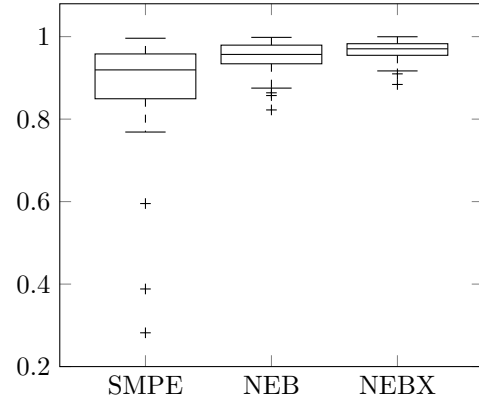| Method	$b_1^0 = 0.2$		$b_2^0 = 0.3$		$a_1^0 = 0.4$		$a_2^0 = 0.5$	
	$E_s\,\hat{b}_1$	$N \cdot \text{Var}_s\,\hat{b}_1$	$E_s\,\hat{b}_2$	$N \cdot \text{Var}_s\,\hat{b}_2$	$E_s\,\hat{a}_1$	$N \cdot \text{Var}_s\,\hat{a}_1$	$E_s\,\hat{a}_2$	$N \cdot \text{Var}_s\,\hat{a}_2$
SMPE	0.20	0.088	0.28	0.075	0.36	1.6	0.53	0.85
NEB	0.21	0.049	0.29	0.070	0.36	0.94	0.52	0.62
NEBX	0.20	0.024	0.29	0.036	0.40	0.60	0.50	0.52

Table 3
Sample mean and sample variance of the parameters estimates for $\hat{G}_{32}$ for the three compared methods.

Method	$b_1^0 = 0.4$		$b_2^0 = 0.5$		$a_1^0 = 0.5$		$a_2^0 = 0.15$	
	$E_s\,\hat{b}_1$	$N \cdot \text{Var}_s\,\hat{b}_1$	$E_s\,\hat{b}_2$	$N \cdot \text{Var}_s\,\hat{b}_2$	$E_s\,\hat{a}_1$	$N \cdot \text{Var}_s\,\hat{a}_1$	$E_s\,\hat{a}_2$	$N \cdot \text{Var}_s\,\hat{a}_2$
SMPE	0.34	1.9	0.44	2.1	0.60	5.0	0.23	3.0
NEB	0.34	0.30	0.44	0.30	0.65	1.0	0.26	0.84
NEBX	0.36	0.11	0.45	0.16	0.63	0.68	0.25	0.55

Fig. 9. Fit of impulse response coefficients of G_{31} for SMPE compared with NEB and NEBX respectively for 100 Monte Carlo simulations. The black line represents $y = x$, i.e., when SMPE performs equally good as NEB and NEBX.

Fig. 10. Box plot of the fit of the impulse response of G_{32} obtained by the methods SMPE, NEB and NEBX respectively.

7 Conclusion

In this paper, we have addressed the identification of a module in dynamic networks with known topology. The problem is cast as the identification of a set of systems in series connection. The second system corresponds to the target module, while the first represents the dynamic relation between exogenous signals and the input and the target module. This system is modeled following a Bayesian kernel-based approach, which enables the identification of the target module using empirical Bayes arguments. In particular, the target module is estimated using a marginal likelihood criterion, whose solution is obtained by a novel iterative scheme designed through the ECM algorithm. The method is extended to incorporate measurements downstream of the target module, which numerical experiments suggest increases performance.
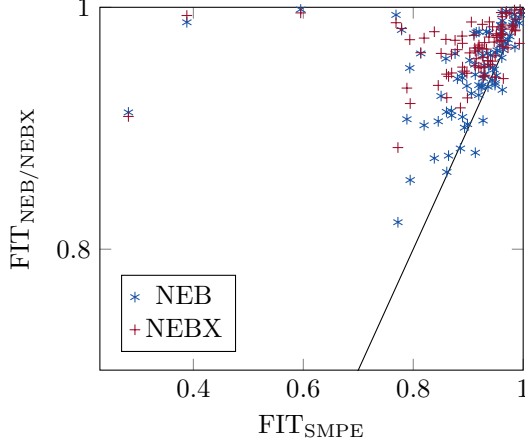
Fig. 11. Fit of impulse response coefficients of G_{32} for SMPE compared with NEB and NEBX respectively for 100 Monte Carlo simulations. The black line represents $y = x$, i.e., when SMPE performs equally good as NEB and NEBX. Note the scaling of the x-axis of this figure.

A   Appendix

Proof of Lemma 4.1

From Bayes' rule it follows that

$$\log p(z, s_{11}; \hat{\eta}^{(k)}) = \log p(z|s_{11}, \ \hat{\eta}^{(k)}) + \log p(s_{11}; \hat{\eta}^{(k)}),$$

with (neglecting constant terms)

$$\log p(z|s_{11}, \ \eta) \propto -\frac{1}{2}\log \det \Sigma_e - \frac{1}{2}\|z - W_\theta s_{11}\|^2_{\Sigma_e^{-1}}$$
$$\log p(s_{11}; \eta) \propto -\frac{1}{2}\log \det \lambda K_\beta - \frac{1}{2}s_{11}^T(\lambda K_\beta)^{-1} s_{11}.$$

Now we have to take the expectation w.r.t. the posterior $p(s_{11}|\tilde{w}_2; \hat{\eta}^{(k)})$. Developing the second term in the first equation above and recalling that

$$\mathrm{E}_{p(s_{11}|\tilde{w}_2; \hat{\eta}^{(k)})}[s_{11}^T A s_{11}] = \mathrm{Tr}\left\{A \hat{S}_{11}^{(k)}\right\},$$

the statement of the lemma readily follows.

Proof of Proposition 4.2

In (26), fix Σ_e to the value $\hat{\Sigma}_e^{(k)}$ (computed inserting $\sigma_1^{2(k)}$ and $\sigma_2^{2(k)}$). We obtain the θ-dependent terms (A.1) and (A.2) (after multiplying by a factor -2), where k_1 and k_2 contain terms independent of θ. Recalling the definitions of $\hat{A}^{(k)}$ and $\hat{b}^{(k)}$, (35) readily follows.

Now, let θ be fixed at the value $\hat{\theta}^{(k+1)}$. The function (26) can be rewritten as (A.3) (after multiplying by a factor -2). The results (36) follow by minimizing (A.3) with respect to σ_1^2 and σ_2^2. Differentiating w.r.t. σ_1^2 and σ_2^2 and calculating the zeros.

Proof of Proposition 5.1

Using Bayes' rule we can decompose the complete likelihood as

$$\begin{aligned}\log p(z, s_{11}, f; \eta) &= \log p(z|s_{11}, f; \eta) \\ &\quad + \log p(s_{11}; \eta) + \log p(f; \eta),\end{aligned}$$

and we will analyze each term in turn. First, note that

$$\begin{aligned}-2\log p(s_{11}|\eta) &= \log \det \lambda_s K_{\beta_s} + s_{11}^T(\lambda_s K_{\beta_s})^{-1} s_{11} \\ &= \log \det \lambda_s K_{\beta_s} + \mathrm{Tr}\left\{(\lambda_s K_{\beta_s})^{-1} s_{11} s_{11}^T\right\}\end{aligned}$$

Replacing $s_{11} s_{11}^T$ with its sample estimate yields the first term in (79). Similarly,

$$-2\log p(f|\eta) = \log \det \lambda_f K_{\beta_f} + \mathrm{Tr}\left\{(\lambda_f K_{\beta_f})^{-1} f f^T\right\}.$$

Replacing $f f^T$ with its sample estimate yields the second term in (79). Finally,

$$-2\log p(z|t, s_{11}; \eta) = \log \det \Sigma + (z - \hat{z})^T \Sigma^{-1}(z - \hat{z}), \tag{A.4}$$

with

$$\hat{z} := \begin{bmatrix} Rs \\ G_\theta Rs \\ G_\theta Rv \end{bmatrix}.$$

The first term of (A.4) is N times the sum of the logarithms of the noise variances squared. The second term of (A.4) decomposes into a sum of the (weighted) error of each signal. Then, the first weighted error is given by

$$\sigma_1^2 \|\tilde{w}_1 - Rs\|_2^2 = \|\tilde{w}_1\|_2^2 - 2\tilde{w}_1^T Rs + \mathrm{Tr}\left\{Rss^T R^T\right\}.$$

Replacing s and ss^T with their respective estimates gives the third term in (79), with the corresponding noise variance term of (A.4) added. Similar calculations on the remaining two weighted errors in (A.4) gives the last two terms in (79). This concludes the proof.

References

Anderson, B. and Moore, J. (1979). *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, N.J., USA.

Bezanson, J., Edelman, A., Karpinski, S. and Shah, V. B. (2017). Julia: A fresh approach to numerical computing, *SIAM Review* **59**(1): 65–98.

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

$$-2z^T \left(\hat{\Sigma}_e^{(k)}\right)^{(-1)} W_\theta \hat{s}_{11}^{(k)} = -\frac{2}{\sigma_2^{2(k)}} y^T G_\theta R_1 \hat{s}_{11}^{(k)} + k_1 \qquad\qquad = -\frac{2}{\sigma_2^{2(k)}} y^T \mathcal{T}_N(R_1 \hat{s}_{11}^{(k)}) g_\theta + k_1 \tag{A.1}$$

$$\mathrm{Tr}\left\{ W_\theta^T \left(\Sigma_e^{(k)}\right)^{-1} W_\theta \hat{S}_{11}^{(k)} \right\} = \frac{1}{\sigma_2^{2(k)}} \mathrm{Tr}\left\{ G_\theta R_1 \hat{S}_{11}^{(k)} R_1^T G_\theta^T \right\} + k_2 = \frac{1}{\sigma_2^{2(k)}} \mathrm{vec}(G_\theta)^T (R_1 \hat{S}_{11}^{(k)} R_1^T \otimes I_N) \mathrm{vec}(G_\theta) + k_2$$

$$= \frac{1}{\sigma_2^{2(k)}} g_\theta^T D^T (R_1 \hat{S}_{11}^{(k)} R_1^T \otimes I_N) D g_\theta + k_2 \, , \tag{A.2}$$

$$Q_0^{(k)}(\sigma_1^2, \sigma_2^2, \hat{\theta}^{(k+1)}) = N(\log \sigma_1^2 + \log \sigma_2^2) + \frac{\|\tilde{w}_1\|_2^2}{\sigma_1^2} + \frac{\|\tilde{w}_2\|_2^2}{\sigma_2^2} - \frac{2\tilde{w}_1^T}{\sigma_1^2} R_1 \hat{s}_{11}^{(k)} - \frac{2\tilde{w}_2^T}{\sigma_2^2} G_{\hat{\theta}^{(k+1)}} R_1 \hat{s}_{11}^{(k)}$$

$$+ \frac{1}{\sigma_1^2} \mathrm{Tr}\left\{ R_1^T R_1 \hat{S}_{11}^{(k)} \right\} + \frac{1}{\sigma_2^2} \mathrm{Tr}\left\{ R_1^T G_{\hat{\theta}^{(k+1)}}^T G_{\hat{\theta}^{(k+1)}} R_1 \hat{S}_{11}^{(k)} \right\} \tag{A.3}$$

Bottegal, G., Risuleo, R. S. and Hjalmarsson, H. (2015). Blind system identification using kernel-based methods, *IFAC-PapersOnLine* **48**(28): 466–471.

Casella, G. (2001). Empirical Bayes Gibbs sampling, *Biostatistics* **2**(4): 485–500.

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

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

Dankers, A. and Van den Hof, P. M. J. (2015). Nonparametric identification in dynamic networks, *Proceedings of the 54th IEEE Conference on Decision and Control*, pp. 3487–3492.

Dankers, A., Van den Hof, P. M. J., Bombois, X. and Heuberger, P. S. (2015). Errors-in-variables identification in dynamic networks - Consistency results for an instrumental variable approach, *Automatica* **62**: 39–50.

Dankers, A., Van den Hof, P. M. J. and Heuberger, P. S. C. (2013). Predictor input selection for direct identification in dynamic networks, *Proceedings of the 52nd IEEE Annual Conference on Decision and Control*, IEEE, pp. 4541–4546.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *J. of the royal statistical society. Series B (methodological)* pp. 1–38.

Everitt, N., Bottegal, G., Rojas, C. R. and Hjalmarsson, H. (2016). Identification of modules in dynamic networks: An empirical bayes approach, *Proceedings of the 55th IEEE Annual Conference on Decision and Control*, IEEE, pp. 4612–4617.

Everitt, N., Bottegal, G., Rojas, C. R. and Hjalmarsson, H. (2017). Variance analysis of linear simo models with spatially correlated noise, *Automatica* **77**: 68–81.

Everitt, N., Galrinho, M. and Hjalmarsson, H. (2016). Optimal model order reduction with the steiglitz-mcbride method, *submitted to Automatica (arXiv:1610.08534)* .

Forssell, U. and Ljung, L. (1999). Closed-loop identification revisited, *Automatica* **35**: 1215–1241.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on pattern analysis and machine intelligence* (6): 721–741.

Gilks, W., Richardson, S. and Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC Interdisciplinary Statistics, Taylor & Francis.

Gilson, M. and Van den Hof, P. M. J. (2005). Instrumental variable methods for closed-loop system identification, *Automatica* **41**(2): 241–249.

Gunes, B., Dankers, A. and Van den Hof, P. M. J. (2014). A variance reduction technique for identification in dynamic networks, *Proceedings of the 19th IFAC World Congress*.

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

Hayden, D., Yuan, Y. and Gonçalves, J. (2014). Network reconstruction from intrinsic noise: Minimum-phase systems, *Proceedings of the 2014 American Control Conference*, pp. 4391–4396.

Hjalmarsson, H. (2009). System identification of complex and structured systems, *European J. of Control* **15**(3-4): 275–310.

Ljung, L. (1998). *System identification*, Springer.

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

Materassi, D. and Salapaka, M. V. (2012). On the problem of reconstructing an unknown topology via locality properties of the Wiener filter, *IEEE Transactions on Automatic Control* **57**(7): 1765–1777.

Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general

framework, *Biometrika* **80**(2): 267–278.

Meyn, S. and Tweedie, R. L. (2009). *Markov chains and stochastic stability; 2nd ed.*, Cambridge Mathematical Library, Cambridge Univ. Press, Leiden.

Pillonetto, G. and De Nicolao, G. (2010). A new kernel-based approach for linear system identification, *Automatica* **46**(1): 81–93.

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

Prando, G., Chiuso, A. and Pillonetto, G. (2014). Bayesian and regularization approaches to multivariable linear system identification: the role of rank penalties, *Proceedings of the 53rd IEEE Annual Conference on Decision and Control*, pp. 1482–1487.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*, The MIT Press.

Risuleo, R. S., Bottegal, G. and Hjalmarsson, H. (2015). A kernel-based approach to Hammerstein system identification, *IFAC-PapersOnLine* **48**(28): 1011–1016.

Torres, P., van Wingerden, J. W. and Verhaegen, M. (2014). Output-error identification of large scale 1D-spatially varying interconnected systems, *IEEE Transactions on Automatic Control* **60**(1): 130–142.

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

Wahlberg, B. (1991). System identification using Laguerre models, *IEEE Transactions on Automatic Control* **36**: 551–562.

Wahlberg, B., Hjalmarsson, H. and Mårtensson, J. (2009). Variance results for identification of cascade systems, *Automatica* **45**(6): 1443–1448.