Version 22-12-2017. Submitted for presentation at 18th IFAC Symposium on System Identification (SYSID 2018), 9-11 July 2018, Stockholm, Sweden.

A sequential least squares algorithm for ARMAX dynamic network identification ⋆

Harm H.M. Weerts * **Miguel Galrinho** ** **Giulio Bottegal** *
Håkan Hjalmarsson ** **Paul M.J. Van den Hof** *

* *Eindhoven University of Technology, Eindhoven, The Netherlands (e-mail: {h.h.m.weerts, g.bottegal, p.m.j.vandenhof}@tue.nl).*
** *Department of Automatic Control, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: {galrinho, hjalmars}@kth.se)*

Abstract: Identification of dynamic networks in prediction error setting often requires the solution of a non-convex optimization problem, which can be difficult to solve especially when dealing with large-scale systems. Focusing on ARMAX models of dynamic networks, in this paper we instead employ a method based on a sequence of least-squares steps. For single-input single-output models, we show that the method is equivalent to Weighted Null Space Fitting and drawing from the analysis of that method, we conjecture that the proposed method is both consistent as well as asymptotically efficient (under suitable assumptions). Simulations indicate that the sequential least squares estimates can be of high quality also for short data sets.

Keywords: System identification, dynamic networks, identification algorithm, least squares

1. INTRODUCTION

Dynamic networks are models of multivariable dynamical systems where, instead of modeling the full system for every input-output relationship, dynamics are modeled locally and interconnected through a particular structure, usually referred to as the network topology. The topology of a dynamic network can be used to gain insights on the physical structure of the system. Furthermore, when the task is to identify the network dynamics from a data set knowing the network topology, this knowledge can be used to decrease the number of free parameters in the model compared to an unstructured input-output model.

Identification of dynamic networks has received considerable attention in recent years. For detection of the network topology, some methods use linear regression models (Sanandaji et al., 2011; Chiuso and Pillonetto, 2012), or the Wiener filter (Materassi and Salapaka, 2012). For identification of a single interconnection modeled as a rational transfer function (referred to as a module), Everitt et al. (2017); Galrinho et al. (2017a) exploit the availability of an external excitation. Identification of a single module is often formulated as a prediction error method (PEM), where one or several multi-input-single-output (MISO) closed-loop identification problems need be solved (Van den Hof et al., 2013; Dankers, 2014; Linder, 2017). To identify the full network, a structured multi-input-multi-output (MIMO) problem is formulated (Weerts et al., 2016, 2017b). The associated PEM cost functions are generally non-convex, and no specific strategy is discussed to solve the optimization problem. Identification methods for dynamic networks that are useful in practice should rely on effective optimization algorithms. This is particularly important for large-scale dynamic networks, where the large number of parameters may lead to high computational complexity and multiple local minima of the cost function.

For networks of arbitrary structure, including loops, with the process noise on each node independent of other noises, the problem complexity can be reduced by splitting the identification procedure into smaller MISO problems. Algorithms that solve the MISO problem are available, e.g. the `ARMAX()` function in Matlab based on Ljung (1999), but also subspace algorithms such as SSARX (Jansson, 2003). However, when process noises are correlated, and there are loops in the network, we cannot split the network dynamics into smaller MISO problems; otherwise, the obtained estimates would be biased. To the best of our knowledge, there are no dedicated algorithms for dynamic network identification with correlated noise sources.

For single-input single-output (SISO) ARMA time-series, Durbin (1960) observed that if the innovations sequence is known, the model can be written as a linear regression model, and the model parameters estimated with least-squares. Based on this, he proposed a method—Durbin's first method—where the innovations are first estimated as the residuals of a high-order AR-model, and then used to estimate the ARMA parameters with least-squares. This method is not asymptotically efficient. The inefficiency can be traced to that the data used in the second step should be filtered with the inverse of the MA-polynomial. As a remedy to this, Mayne and Firoozan (1982) filter the output and the AR-residuals with the inverse of the estimated MA-polynomial obtained from Durbin's first method, and then re-estimate the ARMA-parameters.

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

In this contribution, we extend this method to identification of MIMO ARMAX models, in particular to ARMAX identification of dynamic networks. While the asymptotic results in Mayne and Firoozan (1982) provide a motivation for this type of method, they are not entirely satisfactory in that they do not cover the practical situation where the order of the ARX-model is a function of the sample size. We show the close relation between the proposed method and Weighted Null Space Fitting (WNSF) of Galrinho et al. (2014). Thanks to this we can refer to the thorough asymptotic analysis in Galrinho et al. (2017b).

The paper proceeds as follows. A definition of the dynamic network, of the model, and of the identification method is given (Section 2). The algorithm is formulated (Section 3). A discussion of the algorithm and its estimation properties is given (Section 4). The algorithm is tested in simulations (Section 5), after which the paper is concluded.

2. DYNAMIC NETWORK MODELS

Following the basic setup of Van den Hof et al. (2013), a dynamic network is built up out of L scalar *internal variables* or *nodes* w_j, $j = 1, \dots, L$, and L *external variables* r_j, $j = 1, \cdots L$. Each internal variable is described as:

$$w_j(t) = \sum_{l=1\ (l\neq j)}^{L} G^0_{jl}(q) w_l(t) + \sum_{k=1}^{L} R^0_{jk}(q) r_k(t) + v_j(t) \quad (1)$$

where q^{-1} is the delay operator, i.e. $q^{-1}w_j(t) = w_j(t-1)$;

- G^0_{jl} are strictly proper rational transfer functions, and single transfers G^0_{jl} are referred to as network *modules*
- r_k are *external variables* that can be user-manipulated, and R^0_{jk} are proper rational transfer functions;
- v_j is *process noise*, where the vector process $v = [v_1 \cdots v_L]^T$ is modelled as a stationary stochastic process with rational spectral density, such that there exists a white noise process $e := [e_1 \cdots e_L]^T$ with covariance matrix $\Lambda^0 > 0$ such that $v(t) = H^0(q)e(t)$, with $H^0(q)$ a proper rational transfer function such that $H(\infty) = I$.

When combining the L node signals, we arrive at the network expression (q and t dropped for space limits)

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G^0_{12} & \cdots & G^0_{1L} \\ G^0_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G^0_{L-1\ L} \\ G^0_{L1} & \cdots & G^0_{L\ L-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + R^0 \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_L \end{bmatrix} + H^0 \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_L \end{bmatrix}$$

This results in the matrix equation:

$$w = G^0 w + R^0 r + H^0 e. \quad (2)$$

The open-loop response of the network is denoted by $w(t) = T(q) \begin{bmatrix} e(t) \\ r(t) \end{bmatrix}$, where

$$T(q) = (I - G(q))^{-1} \left[H(q)\ R(q) \right]. \quad (3)$$

To obtain a well-defined network model, we assume:

- $H^0(q)$ is stable and has stable inverse,
- $G^0(q)$ is stable and $(I - G^0(q))$ has stable inverse, and
- $R^0(q)$ is stable and a full rank matrix.

In this paper we focus on MIMO ARMAX model structures. For dynamic networks thses are defined as
$\mathcal{M} = \{M = (D(q,\theta), N_G(q,\theta), N_H(q,\theta), N_R(q,\theta)), \theta \in \Theta\}$, where all matrices are $L \times L$ proper polynomial matrices of order n_p. We consider parameterizations where

- $D(q,\theta)$ is diagonal and monic,
- $N_G(q,\theta)$ has zeros on the diagonal,
- $N_H(q,\theta)$ is monic,
- It is assumed that the model structure is *globally network identifiable at* $M(q,\theta_0)$ (Weerts et al., 2017a).

Network identifiability ensures that the dynamics of the network appear only once in the model set, i.e. $M(q,\theta_0)$ is the only model in $\mathcal{M}$ that describes $M(q,\theta_0)$, which allows to distinguish the contribution of different modules in the network. This is guaranteed when at least one between $H(q,\theta)$ and $R(q,\theta)$ is diagonal (Weerts et al., 2017a).

These matrix polynomials relate to parameterized rational transfer functions

$$G(q,\theta) = D^{-1}(q,\theta) N_G(q,\theta), \quad (4)$$

$$H(q,\theta) = D^{-1}(q,\theta) N_H(q,\theta), \quad (5)$$

$$R(q,\theta) = D^{-1}(q,\theta) N_R(q,\theta). \quad (6)$$

We assume that there exists some parameter θ_0 for which the model captures all dynamics in the network, i.e. $\mathcal{S} = M(\theta_0)$. As D is diagonal, all transfer functions in one row of $[G\ R\ H]$ have the same poles. Note however that, with adequate model orders, this model structure is able to capture any data generating system (1).

2.1 The prediction error criterion

We assume that N samples of $w(t)$ and $r(t)$ are collected for identification. The joint-direct method (Weerts et al., 2017b) is a prediction error-based approach where all nodes are predicted jointly, so that all the network model can be identified simultaneously.

With the ARMAX model defined before, we can define the prediction error of $w(t)$ as

$$\varepsilon(t,\theta) := H^{-1}(q,\theta)\Big((I - G(q,\theta))w(t) - R(q,\theta)r(t)\Big). \quad (7)$$

The identification criterion is then

$$\hat{\theta}_N = \arg\min_{\theta\in\Theta} \frac{1}{N} \sum_{t=1}^{N} \varepsilon^T(t,\theta)\ Q\ \varepsilon(t,\theta), \quad Q > 0, \quad (8)$$

which leads to the model estimate $M(\hat{\theta}_N)$. In Weerts et al. (2017b) it is shown that, under some conditions, (8) is a consistent estimate of the network dynamics, and that the estimate is a maximum likelihood estimate with variance at the Cramér-Rao lower bound.

In general, criterion (8) is a non-convex optimization problem. When the size of the network and the number of modules to be estimated grow, so does the number of local minima. Moreover, computing the direction of the next iterative solution becomes increasingly challenging. In the next section, we propose an estimation method that does not suffer from this problem.

3. SEQUENTIAL LEAST SQUARES

In this section we extend the method in Mayne and Firoozan (1982) from identification of SISO ARMA-models to identification of network ARMAX models.

3.1 Step 1: ARX fitting of network dynamics

This step serves to make an initial estimate of the dynamics, without taking the network structure into account. An ARX model structure is defined as

$$\mathcal{M}_A = \{M_A = (A(q,\eta), B(q,\eta)), \eta \in \beta\},$$

where A and B are $L \times L$ proper polynomial matrices of order n_A, and $A(q,\eta)$ is monic. Using this ARX model a prediction error is defined as

$$\varepsilon_A(t,\eta) = A(q,\eta)w(t) - B(q,\eta)r(t), \tag{9}$$

and optimized with the weighted least squares criterion

$$\hat{\eta}_N = \arg\min_{\eta\in\beta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_A^T(t,\eta)\, Q_A\, \varepsilon_A(t,\eta)\,, \quad Q_A > 0. \tag{10}$$

This estimate has an analytical closed-form solution.

When the ARX estimate is consistent, we also have reconstructed the innovation of the network as $\hat{\varepsilon}_A(t) := \varepsilon_A(t,\hat{\eta}_N)$. The error between the innovation and the reconstructed innovation is denoted by

$$s(t) := e(t) - \hat{\varepsilon}_A(t) \tag{11}$$

and will be used in the later steps of the algorithm.

3.2 Step 2: Reconstructed innovation as input

From (9), we notice that the innovation $e(t)$ can be written as a sum of a known signal $\hat{\varepsilon}_A(t)$ and an unknown signal $s(t)$. Substituting this into (2) yields

$$w = G^0 w + R^0 r + H^0\hat{\varepsilon}_A + H^0 s. \tag{12}$$

Since $\hat{\varepsilon}_A(t)$ is known, it acts as an input in the above network formulation, while s acts as the "new innovation". Note that $s(t)$ becomes smaller when the innovation is estimated better, and if the innovation was recovered exactly, then the above network essentially would be noise-free. When using that $G(q)$, $H(q)$, $R(q)$ share the common denominator $D(q)$, the related prediction error is

$$\varepsilon_s(t,\theta) = N_H^{-1}(q,\theta)L_2(t,\theta), \tag{13}$$

with

$$L_2(t,\theta) = (D(q,\theta) - N_G(q,\theta))w(t) - N_R(q,\theta)r(t) - N_H(q,\theta)\varepsilon_A(t,\hat{\eta}_N). \tag{14}$$

The estimated innovation $\hat{\varepsilon}_A$ acts as an additional input, parameterized with the same parameters as the noise model. The relation between the original prediction error and new prediction error is

$$\varepsilon_s(t,\theta) = \varepsilon(t,\theta) - \hat{\varepsilon}_A, \tag{15}$$

so ε_s is non-linear in the parameters just like ε. However, we note that L_2 is linear in the parameters, and contains the numerator of the noise model N_H. Then, instead of optimizing ε_s, we solve

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} L_2^T(t,\theta)\, Q\, L_2(t,\theta) \quad , Q > 0, \tag{16}$$

which has a closed-form solution in θ. With this criterion we obtain an estimate of all the ARMAX polynomials. In particular, we also obtain an estimate of N_H^{-1}, which we use in the next step of the algorithm. For SISO ARMA models, (16) corresponds to Durbin's first method (Durbin, 1960).

3.3 Step 3: Improve approximation

In step 2, an approximation of $\varepsilon_s(t,\theta)$ is made in order to obtain a linear-in-the parameters criterion that yields an estimate of the parameters. Using the estimate of N_H from step 2, we can construct a new criterion to refine the parameter estimates. To do so, we define a new approximation of $\varepsilon_s(t,\theta)$, where the parameterized term $N_H^{-1}(q,\theta)$ is replaced with the estimated version from the previous step. This can be done for one step, or optionally in an iterative procedure as follows: For $k \geq 3$ use:

$$L_k(t,\theta) := N_H^{-1}(q,\hat{\theta}_N^{[k-1]})L_{k-1}(t,\theta). \tag{17}$$

Then we can define criterion

$$\hat{\theta}_N^{[k]} = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} L_k^T(t,\theta)\, Q\, L_k(t,\theta). \tag{18}$$

The algorithm can be summarized as follows.

Algorithm 1.
1) Choose an ARX model set $\mathcal{M}_A$ with model order n_A, and a parametric network model set $\mathcal{M}$ with model order n.
2) Solve the multivariable linear regression problem (10) with w as the output and r as the input, while using model set $\mathcal{M}_A$.
3) Compute $\hat{\varepsilon}_A = A(\hat{\eta}_N)w - B(\hat{\eta}_N)r$.
4) Solve the linear regression problem (16) where w are the nodes and r and $\hat{\varepsilon}_A$ are the inputs, while using network model set $\mathcal{M}$.
5) Set $k = 3$
6) Invert $N_H(q,\hat{\theta}_N^{k-1})$ to obtain the pre-filter $N_H^{-1}(q,\hat{\theta}_N^{k-1})$.
7) Solve the linear regression problem (18) where the error of step 4) is pre-filtered with $N_H^{-1}(q,\hat{\theta}_N^{k-1})$, while using network model set $\mathcal{M}$.
8) If not converged, increase k by 1, and return to 6). □

4. ANALYSIS AND DISCUSSION

4.1 Relation to standard ARMAX models

A standard MIMO ARMAX model of the form

$$\mathcal{A}(q,\theta)w(t) = \mathcal{B}(q,\theta)r(t) + \mathcal{C}(q,\theta)e(t) \tag{19}$$

can be used to model the dynamics. This ARMAX model is related to the network model through

$$\mathcal{A} = D - N_G, \quad \mathcal{B} = N_R, \quad \mathcal{C} = N_H. \tag{20}$$

Standard ARMAX models and network ARMAX models can be estimated using the same algorithms.

4.2 Connection to WNSF

Although they have been independently derived, in this section we analyze equivalence between Algorithm 1 and WNSF. We now review the basic ideas of WNSF, and refer the reader to Galrinho et al. (2014) for more details.

The first step of WNSF is to estimate a non-parametric ARX model. Then, the parametric model is obtained by equating it to the non-parametric model, using weighted least squares. For example, ARMAX model (19) and the non-parametric ARX model polynomials are related by

$$\begin{cases} \mathcal{C}(q,\theta)A(q,\eta) - \mathcal{A}(q,\theta) = 0 \\ \mathcal{C}(q,\theta)B(q,\eta) - \mathcal{B}(q,\theta) = 0 \end{cases}. \tag{21}$$

Because (21) is linear in θ, we can solve for θ using (weighted) least squares when η is replaced by the available estimate $\hat{\eta}_N$. To do this we write (21) in vector form

$$\hat{\eta}_N - Q(\hat{\eta}_N)\theta =: \varepsilon_\eta(\hat{\eta}_N, \theta), \tag{22}$$

where $Q(\hat{\eta}_N)$ is a block-Toeplitz matrix. These residuals can be used to determine which weighting to use in the weighted least-squares solution. It can be shown that

$$\varepsilon_\eta(\hat{\eta}_N, \theta) \sim As\mathcal{N}\big(0, T_C(\theta)PT_C^\top(\theta)\big). \tag{23}$$

where $\mathcal{N}$ is the Gaussian distribution, $T_C(\theta)$ is a block-Toeplitz matrix function of the parameters in $\mathcal{C}(q,\theta)$, and P is the covariance matrix of the non-parametric least-squares estimate, which can be estimated from data. Then, the optimal weighting to minimize (22) is the inverse of the covariance of the residuals: $W(\theta) = [T_C(\theta)PT_C^\top(\theta)]^{-1}$. Because the weighting depends on θ, we replace it by an estimate, leading to the iterative procedure

$$\hat{\theta}_N^{[k]} = \arg\min \frac{1}{N}\Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big)^\top W(\hat{\theta}_N^{[k-1]})\Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big), \tag{24}$$

Initializing the algorithm can be done, for example, with $W(\hat{\theta}_N^{[0]}) = I$ (the first step is least squares) or with $W(\hat{\theta}_N^{[0]}) = P$ (the first step is with $\hat{\theta}_N^{[0]} = 0$). Any choice of invertible initialization weighting matrix will provide a consistent estimate $\theta_N^{[1]}$. Then, the estimate $\theta_N^{[2]}$ is asymptotically efficient.

A formal proof of equivalence between the methods for a general MIMO ARMAX case is notationally heavy and space consuming; thus, we will not reproduce it here. Instead, the following proposition considers the SISO case.

Proposition 1. Let (19) be a SISO model. Then, the proposed sequential least squares method is, asymptotically in n_A, equivalent to WNSF.

Proof. In the proof, the following notation will be used: let x be a p-dimensional column vector; then, $\mathcal{T}_{n\times m}\{x\}$ $(n \geq p, n \geq m)$ is the $n \times m$ lower-triangular Toeplitz matrix whose first column is $[x^\top \; 0_{1\times n-p}]^\top$. Without loss of generality, we assume the polynomials in (19) are all of order n and there is one delay. Then, for WNSF, we have that (Galrinho et al., 2017b)

$$\begin{aligned}
\eta &= \begin{bmatrix} a_1^\eta & \dots & a_{n_A}^\eta & b_1^\eta & \dots & b_{n_A}^\eta \end{bmatrix}^\top, \\
\theta &= \begin{bmatrix} a_1^\theta & \dots & a_n^\theta & b_1^\theta & \dots & b_n^\theta & c_1^\theta & \dots & c_n^\theta \end{bmatrix}^\top, \\
Q(\theta) &= \begin{bmatrix} \tau_{n_A\times n}(1) & 0 & \tau_{n_A\times n}(\bar{\eta}_{(a)}) \\ 0 & \tau_{n_A\times n}(1) & \tau_{n_A\times n}(\bar{\eta}_{(b)}) \end{bmatrix}, \\
T_C(\theta) &= \begin{bmatrix} \tau_{n_A\times n_A}(\bar{\theta}_{(c)}) & 0 \\ 0 & \tau_{n_A\times n_A}(\bar{\theta}_{(c)}) \end{bmatrix},
\end{aligned} \tag{25}$$

where a_k^η is the coefficient of q^{-k} in $A(q,\eta)$, and analogously for the coefficients of the other polynomials; also, $\bar{\eta}_{(a)} = [1 \; a_1^\eta \; \dots \; a_{n_A-1}^\eta]^\top$, $\bar{\eta}_{(b)} = [1 \; b_1^\eta \; \dots \; b_{n_A-1}^\eta]^\top$, and $\bar{\theta}_{(c)} = [1 \; c_1^\theta \; \dots \; c_n^\theta]^\top$. For the non-parametric covariance, we take $P \propto (\Phi^\top\Phi)^{-1}$ (for SISO, a scaled version of the covariance suffices), where $\Phi = [-\Phi_y \; \Phi_u]$ with

$$\begin{aligned}
\Phi_y &= \tau_{N\times n_A}([u(1) \; \dots \; u(N)]^\top), \\
\Phi_u &= \tau_{N\times n_A}([y(1) \; \dots \; y(N)]^\top).
\end{aligned} \tag{26}$$

In (26) we assumed zero initial conditions, but it does not have to be the case (this is discussed in Section 5). With these definitions, we may write the WNSF estimate as

$$\hat{\theta}_N^{[k]} = \arg\min \frac{1}{N}\mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta)^\top \mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta) \tag{27}$$

where

$$\mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta) = \Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big). \tag{28}$$

To observe equivalence between the methods, we must write (27) in a form similar to (18) (i.e., using filters). Indeed, the Toeplitz structure of the matrices in (28) provides a filtering interpretation. Taking one term from (28), then entry t of the vector can alternatively be expressed in filtering form as

$$[\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\hat{\eta}_N]_t \approx \frac{B(q,\hat{\eta}_N)}{C(q,\hat{\theta}_N^{[k-1]})}u(t) - \frac{A(q,\hat{\eta}_N)-1}{C(q,\hat{\theta}_N^{[k-1]})}y(t) \tag{29}$$

and similarly for $\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})Q(\hat{\eta}_N)\theta$. The only difference (and therefore the approximation sign) is in the tail of the filters because of the way that the filters are truncated in matrix form. This difference was treated in detail by Galrinho (2016) for OE models; it is negligible in practice, and it vanishes asymptotically in n_A.

Then, we have that, asymptotically in n_A, the WNSF estimate (24) can be written as

$$\begin{aligned}
\hat{\theta}_N^{[k]} = \arg\min \frac{1}{N}\sum_{t=1}^{N} \big(C^{-1}(q,\hat{\theta}_N^{[k-1]})\,\{[B(q,\hat{\eta}_N)\,\mathcal{C}(q,\theta) \\
-\mathcal{B}(q,\theta)]u_t - [\mathcal{A}(q,\theta) - A(q,\hat{\eta}_N)\mathcal{C}(q,\theta)]y_t\}\big)^2.
\end{aligned} \tag{30}$$

Using (9), (14), and (17), (18) can be written in the same form as (30), which concludes the proof. □

4.3 Asymptotic properties

Although informal, the following argument provides an intuition for consistency of Algorithm 1. When a consistent ARX estimate is obtained, then the prediction error converges to the innovation, i.e. $\varepsilon_A(t,\hat{\eta}_N) \to e(t)$. The estimated innovation sequence is used as a known input in the ARMAX estimation step (16), and the $s(t)$ acts as the "new innovation". We can investigate whether θ_0 is a minimum of (16). When substituting the true network into L_2, we obtain the expression

$$\begin{aligned}
L_2(t,\theta) = &\Big(X(\theta)N_R(\theta_0) - N_R(\theta)\Big)r(t) \\
&+ \Big(X(\theta)N_H(\theta_0) - N_H(\theta)\Big)\hat{\varepsilon}_A(t) \\
&+ \; X(\theta)N_H(\theta_0)s(t),
\end{aligned} \tag{31}$$

driven only by external signals, with

$$X(\theta) = (D(\theta) - N_G(\theta))(D(\theta_0) - N_G(\theta_0))^{-1}. \tag{32}$$

We can see that for $\theta = \theta_0$ the $X = 1$ and then the first 2 terms of (31) are 0, and so $L_2(t,\theta_0) = N_H(\theta_0)s(t)$. Due to consistency of the ARX estimate,

$$s(t) = e(t) - \varepsilon_A(t,\hat{\eta}_N) \to 0, \tag{33}$$

which implies that $L_2(t,\theta_0) \to 0$. Then, the cost function of (16) is 0 and minimized by θ_0.

Given that the proposed algorithm is asymptotically the same as WNSF, consistency and asymptotic efficiency follow from extending the results by Galrinho et al. (2017b) to MIMO ARMAX models. Doing this extension in detail would be notationally heavy, but it is easy to understand

that the results therein still apply in the MIMO case. The challenge in proving consistency and asymptotic efficiency is to keep track of the bias and variance errors in the non-parametric model. It is possible that n_A must tend to infinity to obtain unbiased estimates: for that, we make n_A a function of sample size N, as suggested by Ljung and Wahlberg (1992). Assuming standard identifiability conditions for MIMO systems (Ljung, 1999), the key step in extending the proof of Galrinho et al. (2017b) is to guarantee that the difference between the noisy weighted regression matrices and the true ones converge in some norm. In the MIMO case, these matrices will be larger, but their dimensional increase still satisfies the same rates. Thus, assuming standard rates of increase of $n(N)$ (Ljung and Wahlberg, 1992)—$n(N)^{4+\delta}/N \to 0$ ($\delta > 0$) suffices for all theoretical results—the method is consistent and asymptotically efficient.

5. IMPLEMENTATION ISSUES AND SIMULATIONS

In this section, simulations are performed to validate the theoretical results, and to see how the models estimated by Algorithm 1 compare to the PEM estimate (8). However, first we discuss some implementation aspects.

5.1 Selecting the ARX-order and the number of iterations

The quality of the estimated parametric may be influenced by the order of the ARX model. Since it is known that PEM is asymptotically efficient, one possibility is to choose the ARX order which minimizes (8) (Galrinho et al., 2017b). The same criterion can be used to select the number of iterations. If some combinations of ARX order and iteration lead to unstable models for particular data sets, these unstable models will not be selected.

5.2 Transient estimation

With the ARX-model being of high order, the handling of transient effects becomes important for short data records. Estimating the transient can then improve the model quality since the transient contains information regarding the poles of the network. We will not go into details here on how this can be done but refer to Galrinho et al. (2015).

5.3 A SISO model

The network used for showing the asymptotic and iterative properties is depicted in Fig. 1 for $L = 3$, i.e. the network consists of 3 nodes. We try to estimate G_{32} from data. This network has these dynamics

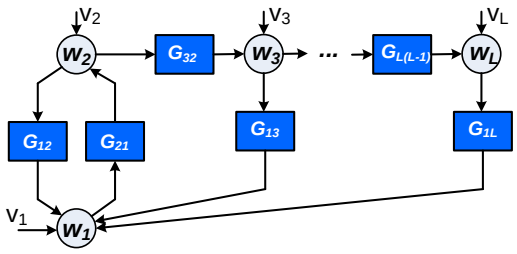

Fig. 1. An L node network used for simulations.

$$G_{32} = \frac{q^{-1} + 0.5q^{-2}}{1 - 0.5q^{-1} + 0.2q^{-2}}, \qquad H_{33} = \frac{1 - 0.6q^{-1}}{1 - 0.85q^{-1}},$$
$$G_{21} = \frac{0.4q^{-1} - 0.2q^{-2}}{1 + 0.4q^{-1} - 0.5q^{-2}}, \qquad H_{22} = \frac{1 - 0.3q^{-1}}{1 - 0.9q^{-1}},$$
$$G_{13} = \frac{0.8q^{-1}}{1 - 0.3q^{-1}}, \qquad H_{11} = \frac{1 + 0.5q^{-1}}{1 - 0.7q^{-1}},$$
$$G_{12} = \frac{-0.7q^{-1}}{1 - 0.7q^{-1}}.$$

The Gaussian white noises driving the network are uncorrelated with equal power. Since H is diagonal, the joint-direct method consists of 3 separate MISO problems that can individually be solved. A 3rd order ARMAX model is estimated for node 3. Mean squared error of the impulse response is used as the performance measure.

Simulations with varying number of samples N are performed over 100 Monte-Carlo runs, for the sufficiently high ARX order $n_A = 35$. For each different data length N, 100 Monte-Carlo runs are performed, and the MSE of module $G_{32}(\hat{\theta}_N)$ is averaged over the runs. Fig. 2 shows the resulting average MSE per data length N for the 2nd, 3rd and 10th iteration of Algorithm 1, and for the joint-direct estimate (8) computed by the Matlab algorithm `armax()` using the true system as initialization.

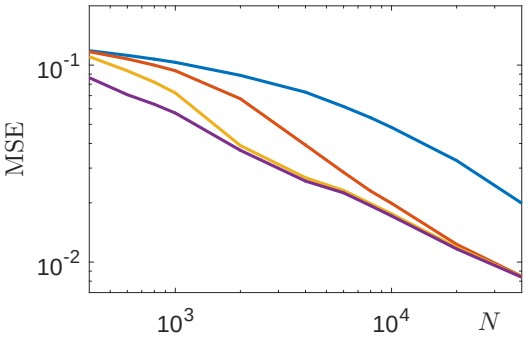

Fig. 2. Average MSE over 100 Monte-Carlo runs plotted against data length N. Blue: Step 2, Red: Step 3, Yellow: 10-th iteration, and Purple: the direct method.

For small number of data N, Algorithm 1 has a slightly higher MSE than PEM. Increasing N leads to improved models for each step of the algorithm. Around $N = 3 \cdot 10^4$, step 3 has the same MSE as PEM, and around $N = 6 \cdot 10^3$ the 10-th iteration has the same MSE as PEM. The point of these simulations is to show that this algorithm with explicit solution is a close approximation of the non-convex optimization problem (8), even for small data sets.

5.4 A 5 node network

Estimation of a more challenging 5 node network with randomly generated dynamics will now be tested with a fixed number of samples $N = 1000$. We try to estimate G_{12} in a MISO setting. The network is shown in Fig. 1 for $L = 5$. In each Monte-Carlo run, the modules are randomly generated with restrictions:

- Modules are randomly generated by `drss()` and of 2nd order, with $|G(z)| < 0.9$ and and $\|G(z)\|_{\mathcal{H}_2} = 0.5$.
- The closed-loop transfer and the predictor filters satisfy $|T(z)| < 0.95$ and $|W(z)| < 0.95$ respectively.

Process noises are colored, H_i is first order, and all driving white noises are uncorrelated with equal power. Models are estimated with:

- Algorithm 1 with order and iteration selection as specified in Section 5.1.
- The SSARX subspace identification algorithm implemented in Matlab as part of the `n4sid()` function.
- PEM, with the `armax()` algorithm of Matlab with standard initialization.
- PEM, with the `armax()` algorithm of Matlab with the true system as initialization.

Performance is evaluated by fit ratio defined by

$$f_g(\theta) = 1 - \frac{\|g(\theta_0) - g(\theta)\|_2}{\|g(\theta_0)\|_2}, \tag{34}$$

where g is the impulse response of a module. In total 100 Monte-Carlo runs are performed, and the resulting fit of module G_{12} is shown in Fig. 3. The PEM algorithm starts to struggle with these 4 inputs, but overall the performance of the 4 algorithms is competitive. We conclude that Algorithm 1 is suitable for extension to MISO and MIMO, and therefore suitable for use in dynamic networks.

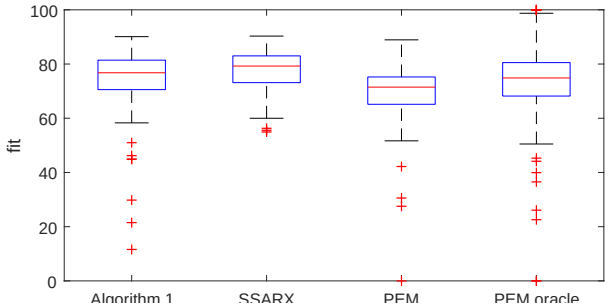

Fig. 3. Fit of G_{12} for 100 randomly generated systems and data sets for: Algorithm 1, SSARX, PEM with standard initialization, PEM initialized by true system.

6. CONCLUSIONS

We have presented a method for identification of dynamic networks of ARMAX-type that is based on a sequence of least-squares steps. It generalizes the method for ARMA models presented in Mayne and Firoozan (1982) and we have shown that for SISO models it corresponds to WNSF, which through the analysis in (Galrinho et al., 2017b) suggests that consistency and asymptotic efficiency can be established under suitable conditions on the rate of increase of the ARX-model order as function of the sample size. With simulations we have verified that the method performs comparable to PEM, i.e. that it achieves asymptotic efficiency, as the sample size grows in a simple SISO setting subject to Gaussian disturbances. Further, in a more challenging network setting, the method was shown to be competitive with PEM, even when this algorithm was initialized at the true parameter values. This means that in the future this algorithm can be used for example for topology detection.

REFERENCES

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

Dankers, A.G. (2014). *System identification in dynamic networks*. Ph.D. thesis, Delft University of Technology.

Durbin, J. (1960). The fitting of time-series models. *Revue de l'Institut International de Statistique*, 28(3), 233–244.

Everitt, N., Bottegal, G., and Hjalmarsson, H. (2017). An empirical Bayes approach to identification of modules in dynamic networks. *Automatica*. To appear.

Galrinho, M. (2016). *Least Squares Methods for System Identification of Structured Models*. Licentiate thesis, KTH Royal Institute of Technology.

Galrinho, M., Everitt, N., and Hjalmarsson, H. (2017a). Incorporating noise modeling in dynamic networks using non-parametric models. In *20th IFAC World Congress*, 10568–10573. Toulouse, France.

Galrinho, M., Rojas, C., and Hjalmarsson, H. (2014). A weighted least-squares method for parameter estimation in structured models. In *53rd IEEE Conference on Decision and Control*, 3322–3327. Los Angeles, USA.

Galrinho, M., Rojas, C., and Hjalmarsson, H. (2015). On estimating initial conditions in unstructured models. In *54th IEEE Conference on Decision and Control*, 2725–2730. Osaka, Japan.

Galrinho, M., Rojas, C., and Hjalmarsson, H. (2017b). Parametric identification using weighted null-space fitting. *arXiv preprint arXiv:1708.03946*.

Jansson, M. (2003). Subspace identification and arx modeling. In *13th IFAC Symp on System Identification, Rotterdam, The Netherlands*.

Linder, J. (2017). *Indirect system identification for unknown input problems with applications to ships*. PhD dissertation, Linköping University.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ.

Ljung, L. and Wahlberg, B. (1992). Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra. *Advances in Applied Probability*, 24(2), 412–440.

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

Mayne, D. and Firoozan, F. (1982). Linear identification of ARMA processes. *Automatica*, 18, 461–466.

Sanandaji, B.M., Vincent, T.L., and Wakin, M.B. (2011). Exact topology identification of large-scale interconnected dynamical systems from compressive observations. In *American Control Conference*, 649–656. San Francisco, CA, USA.

Van den Hof, P.M.J., Dankers, A.G., 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.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2016). Identification of dynamic networks operating in the presence of algebraic loops. In *55nd IEEE Conference on Decision and Control*, 4606–4611.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2017a). Identifiability of linear dynamic networks. *Automatica*. (To appear) arXiv preprint arXiv:1609.00864.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2017b). Identification of linear dynamic networks with rank-reduced noise. *arXiv preprint arXiv:1711.06369*.