To appear in Proc. 18th IFAC Symposium on System Identification, 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 for large-scale systems. Focusing on ARMAX models of dynamic networks, 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 the recently developed 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 even for short data sets.

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

1. INTRODUCTION

Dynamic networks are models of multivariable 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. (2018); 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, 2017). 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. The main question is whether there are algorithms suitable for PEM identification of large-scale dynamic networks.

The network identification problem is often split into a smaller MISO problem for each node. 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, identification of the global network in a MIMO setting does lead to consistent estimates; however, this comes at the cost of increasing computational complexity. Subspace methods are usually suitable for large-scale settings, but it is not possible to encode the topology into e.g. SSARX.

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 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, which was remedied in Mayne and Firoozan (1982) by filtering the output and the AR-residuals with the inverse of the estimated MA-polynomial, and then re-estimate the ARMA-parameters. The asymptotic results in Mayne and Firoozan (1982) are not entirely satisfactory, since they do not cover the situation where the order of

⋆ 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).

the ARX model is a function of the sample size, which is addressed in Hannan and Kavalieris (1983). This type of method has become popular for vector ARMA time series (e.g., Dufour and Jouini (2014) and references therein) due to their computational and optimal statistical properties.

We propose an approach for the identification of ARMAX models in dynamic networks that belongs to that class of identification methods. This is especially important when process noises are correlated: the flexibility of the method allows to encode the network topology and capture the noise correlations. In order to illustrate the asymptotic properties, we show the close relation between the proposed method and Weighted Null Space Fitting (WNSF) of Galrinho et al. (2014), such that we can refer to the thorough asymptotic analysis in Galrinho et al. (2017b).

The paper proceeds proceeds with a network definition (Section 2), a formulation of the algorithm (Section 3), a discussion of the algorithm and its estimation properties (Section 4), and finally simulations (Section 5).

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 modeled 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:

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

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 defined by parameterized rational transfer functions $G(q,\theta) = D^{-1}(q,\theta)N_G(q,\theta)$, $H(q,\theta) = D^{-1}(q,\theta)N_H(q,\theta)$, $R(q,\theta) = D^{-1}(q,\theta)N_R(q,\theta)$, where all matrices are $L \times L$ proper polynomial matrices of order n_p and 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.

The related model set is defined as

$$\mathcal{M} = \{M = (D(q,\theta), N_G(q,\theta), N_H(q,\theta), N_R(q,\theta)), \theta \in \Theta\}.$$

It is assumed that the model structure is *globally network identifiable at* $M(q,\theta_0)$ (Weerts et al., 2018). 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., 2018).

As D is diagonal, all transfer functions in one row of $[G\ R\ H]$ have the same poles. With adequate model orders this model structure captures any network (2) due to pole-zero cancellations. There exists a θ_0 such that $\mathcal{S} = M(\theta_0)$.

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., 2017) is a prediction error-based approach where all nodes are predicted jointly, such that the global network model is identified. The prediction error is defined by

$$\varepsilon(t,\theta) := H^{-1}(q,\theta)\Big((I - G(q,\theta))w(t) - R(q,\theta)r(t)\Big). \quad (3)$$

The identification criterion is then

$$\hat{\theta}_N = \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon^T(t,\theta)\, Q\, \varepsilon(t,\theta), \quad Q > 0, \quad (4)$$

which leads to the model estimate $M(\hat{\theta}_N)$. In Weerts et al. (2017) it is shown that, under some conditions, (4) is a consistent estimate, which is a maximum likelihood estimate with variance at the Cramér-Rao lower bound.

In general, criterion (4) is a non-convex optimization problem, and the number of local minima grows with the number of modules to be estimated.

3. SEQUENTIAL LEAST SQUARES

In this section we introduce the developed algorithm.

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

$$\varepsilon_A(t,\eta) = A(q,\eta)w(t) - B(q,\eta)r(t), \quad (5)$$

and optimized with the weighted least squares criterion

$$\hat{\eta}_N = \arg\min_{\eta\in\beta} \frac{1}{N}\sum_{t=1}^{N} \varepsilon_A^T(t,\eta)\, Q_A\, \varepsilon_A(t,\eta), \quad Q_A > 0. \quad (6)$$

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

$$s(t) := e(t) - \hat{\varepsilon}_A(t) \tag{7}$$

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

3.2 Step 2: Reconstructed innovation as input

From (7), 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

$$w = G^0 w + R^0 r + H^0 \hat{\varepsilon}_A + H^0 s. \tag{8}$$

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

$$\varepsilon_s(t, \theta) = N_H^{-1}(q, \theta) L_2(t, \theta), \tag{9}$$

with

$$\begin{aligned} L_2(t, \theta) = (D(q, \theta) - N_G(q, \theta))w(t) \\ - N_R(q, \theta) r(t) - N_H(q, \theta) \varepsilon_A(t, \hat{\eta}_N). \end{aligned} \tag{10}$$

Note that D is diagonal and N_G is 0 on the diagonal, so that their parameterizations do not mix. The estimated innovation $\hat{\varepsilon}_A$ acts as an additional input, parameterized with the same parameters as the noise model. The new prediction error ε_s is non-linear in the parameters, similarly to 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

$$\hat{\theta}_N^{[2]} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{t=1}^{N} L_2^T(t, \theta)\, Q\, L_2(t, \theta) \quad , Q > 0, \tag{11}$$

which has a closed-form solution in θ. With this criterion we obtain an estimate of all the 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, (11) corresponds to Durbin's first method (Durbin, 1960).

3.3 Step 3: Improve approximation

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

$$L_k(t, \theta) := N_H^{-1}(q, \hat{\theta}_N^{[k-1]}) L_{k-1}(t, \theta). \tag{12}$$

Then we can define criterion

$$\hat{\theta}_N^{[k]} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{t=1}^{N} L_k^T(t, \theta)\, Q\, L_k(t, \theta). \tag{13}$$

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 (6) 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 (11) 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 pre-filter $N_H^{-1}(q, \hat{\theta}_N^{k-1})$.
7) Solve the linear regression problem (13) 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

$$\mathcal{A}(q, \theta) w(t) = \mathcal{B}(q, \theta) r(t) + \mathcal{C}(q, \theta) e(t) \tag{14}$$

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

$$\mathcal{A} = D - N_G, \quad \mathcal{B} = N_R, \quad \mathcal{C} = N_H. \tag{15}$$

Network ARMAX models can be estimated using standard ARMAX algorithms if the topology can be encoded. In general, this cannot be done with the ARMAX function in MATLAB, which is restricted to a diagonal C matrix.

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 (14) and the non-parametric ARX model polynomials are related by

$$\begin{cases} \mathcal{C}(q, \theta) A(q, \eta) - \mathcal{A}(q, \theta) = 0 \\ \mathcal{C}(q, \theta) B(q, \eta) - \mathcal{B}(q, \theta) = 0 \end{cases}. \tag{16}$$

Because (16) 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 (16) in vector form

$$\hat{\eta}_N - Q(\hat{\eta}_N)\theta =: \varepsilon_\eta(\hat{\eta}_N, \theta), \tag{17}$$

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

$$\varepsilon_\eta(\hat{\eta}_N, \theta) \sim As\mathcal{N}\big(0, T_C(\theta) P T_C^\top(\theta)\big). \tag{18}$$

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 (17) is the inverse of the covariance of the residuals: $W(\theta) = [T_C(\theta) P T_C^\top(\theta)]^{-1}$. Because the weighting depends on θ, we replace it by an estimate, leading to the iterative procedure

$$\hat{\theta}_N^{[k]} = \arg\min \frac{1}{N} \Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big)^\top W(\hat{\theta}_N^{[k-1]}) \Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big), \tag{19}$$

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, we will in this exposition use the SISO case to provide the intuition on why the methods are equivalent. Without loss of generality, we assume the polynomials in (14) are all of order n and there is one delay. 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 $n \times m$ Toeplitz matrix with first column $[x^\top \ 0_{1\times n-p}]^\top$ and first row $[x_1 \ 0_{1\times m-1}]$, where x_1 is the first element of x.

We start by defining the matrices in (17) and (18) used by WNSF in the SISO ARMAX case (we refer to Galrinho et al. (2017b) for more details), for which we have

$$\begin{aligned}
\eta &= \begin{bmatrix} a_1^\eta & \dots & a_{n_A}^\eta & b_1^\eta & \dots & b_{n_A}^\eta \end{bmatrix}^\top, \\
\theta &= \begin{bmatrix} a_1^\theta & \dots & a_n^\theta & b_1^\theta & \dots & b_n^\theta & c_1^\theta & \dots & c_n^\theta \end{bmatrix}^\top, \\
Q(\eta) &= \begin{bmatrix} \tau_{n_A\times n}(1) & 0 & \tau_{n_A\times n}(\bar{\eta}_{(a)}) \\ 0 & \tau_{n_A\times n}(1) & \tau_{n_A\times n}(\bar{\eta}_{(b)}) \end{bmatrix}, \\
T_C(\theta) &= \begin{bmatrix} \tau_{n_A\times n_A}(\bar{\theta}_{(c)}) & 0 \\ 0 & \tau_{n_A\times n_A}(\bar{\theta}_{(c)}) \end{bmatrix},
\end{aligned} \tag{20}$$

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

$$\begin{aligned}
\Phi_y &= \tau_{N\times n_A}([u(1) \ \dots \ u(N)]^\top), \\
\Phi_u &= \tau_{N\times n_A}([y(1) \ \dots \ y(N)]^\top).
\end{aligned} \tag{21}$$

In (21) 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

$$\hat{\theta}_N^{[k]} = \arg\min \frac{1}{N}\mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta)^\top \mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta), \tag{22}$$

where

$$\mathcal{L}(\hat{\eta}_N, \hat{\theta}_N^{[k-1]}; \theta) = \Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\Big(\hat{\eta}_N - Q(\hat{\eta}_N)\theta\Big). \tag{23}$$

To observe equivalence between the methods, we must write (22) in a form similar to (13) (i.e., using filters). Indeed, the Toeplitz structure of the matrices in (23) provides a filtering interpretation. Consider, for example, the term $\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\hat{\eta}_N$: it corresponds to filtering the data in Φ (i.e., the inputs and outputs) by the product of the inverse of $C(q,\hat{\theta}_N^{[k-1]})$ with the non-parametric estimates $\hat{\eta}_N$ of the ARX model polynomials. However, because the matrices have dimension n_A, the filters are truncated to n_A coefficients. This difference was treated by Galrinho (2016) for OE models, and it is negligible in practice for a reasonably large n_A. In the ARMAX case, it is analogous: the t^{th} element of $\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\hat{\eta}_N$ can be written as

$$[\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\hat{\eta}_N]_t = \sum_{k=1}^{n_A}\beta_k q^{-k}u(t) - \sum_{k=1}^{n_A}\alpha_k q^{-k}y(t), \tag{24}$$

where

$$\sum_{k=1}^{\infty}\alpha_k q^{-k} := \frac{A(q,\hat{\eta}_N)-1}{\mathcal{C}(q,\hat{\theta}_N^{[k-1]})}, \quad \sum_{k=1}^{\infty}\beta_k q^{-k} := \frac{B(q,\hat{\eta}_N)}{\mathcal{C}(q,\hat{\theta}_N^{[k-1]})}. \tag{25}$$

Because $A(q,\hat{\eta}_N)$ and $B(q,\hat{\eta}_N)$ are already of order n_A, the main effect of truncating the filters (25) according to (24) is that we truncate the tail of the impulse response of $1/C(q,\hat{\theta}_N^{[k-1]})$. This has a negligible effect for n_A large enough. Hence, we may write approximately

$$[\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})\hat{\eta}_N]_t \approx \frac{B(q,\hat{\eta}_N)}{\mathcal{C}(q,\hat{\theta}_N^{[k-1]})}u(t) - \frac{A(q,\hat{\eta}_N)-1}{\mathcal{C}(q,\hat{\theta}_N^{[k-1]})}y(t). \tag{26}$$

Using a similar approach for the remaining term in (22), given by $\Phi T_C^{-1}(\hat{\theta}_N^{[k-1]})Q(\hat{\eta}_N)\theta$, we have that the WNSF estimate (19) can be approximately written as

$$\begin{aligned}
\hat{\theta}_N^{[k]} \approx \arg\min \frac{1}{N}\sum_{t=1}^{N}\big(\mathcal{C}^{-1}(q,\hat{\theta}_N^{[k-1]})\{[B(q,\hat{\eta}_N)\,\mathcal{C}(q,\theta) \\
-\mathcal{B}(q,\theta)]u_t - [\mathcal{A}(q,\theta) - A(q,\hat{\eta}_N)\mathcal{C}(q,\theta)]y_t\}\big)^2,
\end{aligned} \tag{27}$$

with the approximation error diminishing for larger n_A. Using (5), (10), and (12), (13) can be written in the same form as (27), which shows the equivalence of the methods.

Equivalence of the methods has two implications. Algorithm 1 is an alternative interpretation of WNSF. And then in the SISO case, Step 2 of SLS is a consistent estimate, and Step 3 is an asymptotically efficient estimate.

4.3 Asymptotic properties

Although informal, the following argument provides an intuition for consistency of Algorithm 1 in MIMO setting. When the ARX estimate is consistent, 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 (11), and the $s(t)$ acts as the "new innovation". We can investigate whether θ_0 is a minimum of (11). When substituting the true network into L_2, we obtain the expression

$$\begin{aligned}
L_2(t,\theta) &= \Big(X(\theta)N_R(\theta_0) - N_R(\theta)\Big)r(t)\cdot \\
&\cdot + \Big(X(\theta)N_H(\theta_0) - N_H(\theta)\Big)\hat{\varepsilon}_A(t) + X(\theta)N_H(\theta_0)s(t),
\end{aligned} \tag{28}$$

driven only by external signals, with

$$X(\theta) = (D(\theta) - N_G(\theta))(D(\theta_0) - N_G(\theta_0))^{-1}. \tag{29}$$

We can see that for $\theta = \theta_0$ the $X = 1$ and then the first 2 terms of (28) are 0, and so $L_2(t,\theta_0) = N_H(\theta_0)s(t)$. Due to consistency of the ARX estimate,

$$s(t) = e(t) - \varepsilon_A(t,\hat{\eta}_N) \to 0, \tag{30}$$

which implies that $L_2(t,\theta_0) \to 0$. Then, the cost function of (11) 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

5.1 Selecting the ARX-order and the number of iterations

The quality of the estimated parameters 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 (4) (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

Asymptotic and iterative properties will be shown by estimating G_{32} in the network depicted in Fig. 1 for $L = 3$. The network consists of 3 nodes with these dynamics

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

The Gaussian white noises driving the network are uncorrelated and of equal power. Since H is diagonal, the joint-direct method consists of three separate MISO problems that can individually be solved. A third 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

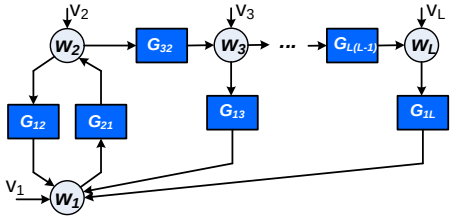

Fig. 1. An L node network used for simulations.

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 (4) computed by the Matlab algorithm `armax()` using the true system as initialization.

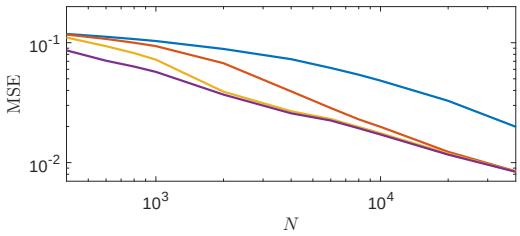

Fig. 2. Average MSE over 100 Monte-Carlo runs plotted against data length N. Blue: Step 2, Red: Step 3, Yellow: 10-th iteration, and Purple: the direct method.

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 (4), even for small data sets.

The WNSF algorithm, with weight $W(\hat{\theta}_N^{[0]}) = P$ in the second step, was tested on the same data, resulting in a plot indistinguishable from Fig. 2, which would be redundant to include.

5.4 A 5 node network

In order to show that the algorithm is competitive in a multi-input setting we include an estimation of the network in Fig. 1 for $L = 5$ with randomly generated dynamics. Module G_{12} is estimated in a 4-input-1-output setting using samples $N = 1000$ samples. Per 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 T and the predictor filters W 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. A network ARMAX model of order 9 is able to exactly model the network. Models are estimated with these algorithms:

- 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

$$f_g(\theta) = 1 - \frac{\|g(\theta_0) - g(\theta)\|_2}{\|g(\theta_0)\|_2}, \tag{31}$$

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 the performance of each algorithm is competitive. For the MISO case, Algorithm 1 is competitive with existing algorithms.

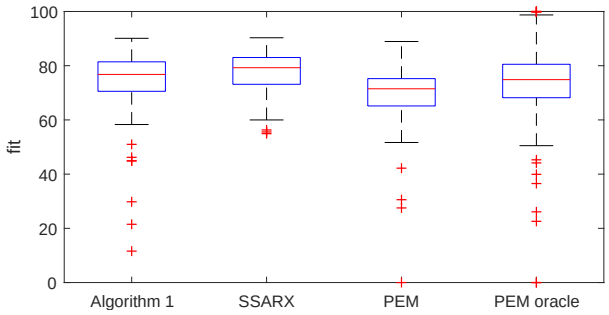

Fig. 3. Fit of G_{12} for 100 randomly generated systems and data sets for: Algorithm 1, SSARX, PEM with standard initialization, PEM initialized by true system.

6. CONCLUSIONS

We have presented a method for identification of dynamic networks 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 achieves asymptotic efficiency in a SISO setting subject to Gaussian disturbances. In a more challenging network setting, the method is competitive with PEM and SSARX. because it can be straightforwardly applied with MIMO ARMAX (where SSARX cannot encode the network topology), we conclude that Algorithm 1 is promising for global network identification with correlated noise, also with the potential use of providing initialization points for PEM.

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.

Dufour, J.M. and Jouini, T. (2014). Asymptotic distributions for quasi-efficient estimators in echelon varma models. *Computational Statistics & Data Analysis*, 73, 69–86.

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. (2018). An empirical Bayes approach to identification of modules in dynamic networks. *Automatica*, 91, 144 – 151.

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*.

Hannan, E.J. and Kavalieris, L. (1983). Linear estimation of arma processes. *Automatica*, 19(4), 447–448.

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. (2017). Identification of linear dynamic networks with rank-reduced noise. *arXiv preprint arXiv:1711.06369*.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2018). Identifiability of linear dynamic networks. *Automatica*, 89, 247–258.