



Delft University of Technology

Identification of 2D Interconnected Systems An efficient Steepest-Descent approach

Torres, Patricio; Wingerden, Jan Willem van

DOI

[10.1016/j.ifacol.2018.09.094](https://doi.org/10.1016/j.ifacol.2018.09.094)

Publication date

2018

Document Version

Final published version

Published in

IFAC-PapersOnLine

Citation (APA)

Torres, P., & Wingerden, J. W. V. (2018). Identification of 2D Interconnected Systems: An efficient Steepest-Descent approach. *IFAC-PapersOnLine*, 51(15), 78-83.
<https://doi.org/10.1016/j.ifacol.2018.09.094>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Identification of 2D Interconnected Systems: An efficient Steepest-Descent approach.

Patricio Torres* and Jan-Willem van Wingerden**

* Center for Mathematical Modeling, UMI 2807 CNRS, University of Chile, Box 170/3, Mail 3, Santiago, Chile (Tel: +56 22 557 8963; e-mail: pattorre@gmail.com).

** Delft University of Technology, Delft, 2628 CD The Netherlands (Tel: +3115 27 81720; e-mail: J.W.vanWingerden@tudelft.nl).

Abstract: In this paper, a new identification method for large 2D grids of interconnected systems is presented. The proposed algorithm minimizes the Output-Error of the lifted system by using a Steepest-Descent optimization method which exploit the Multilevel Sequentially-Semi Separable (MSSS) structure of the involved matrices. Furthermore, it is shown that the computational complexity of the proposed approach is linear with respect to the number of subsystems in the grid. Finally, a numerical example is presented in order to show the effectiveness of the proposed algorithm.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Interconnected systems, Output-Error identification, Steepest-Descent, Jacobian, Efficient Algorithms.

1. INTRODUCTION

In recent years, there has been an increasing interest in identification and modeling of interconnected systems. This has been motivated for the large number of applications where interconnected systems appear, such as: car platoons, Horowitz and Varaiya (2000), structural beams, Scholte and D'Andrea (2003), irrigation networks, Cantoni et al. (2007), turbulent wavefront reconstruction, Fraanje et al. (2010), satellite formations, Massioni (2010) between others. In the case of Prediction-Error methods (PEM), an identification approach for Linear Parameter Varying (LPV) spatially interconnected systems in input-output form based on Least Squares (LS) estimation is presented in Mukhtar et al. (2010). This approach has also been extended to Box-Jenkins models in Mukhtar et al. (2011). In Dankers (2015), the theory of closed-loop system identification is extended to networks of dynamical systems with known topology. Moreover, Weerts et al. (2016) studies the conditions for uniquely identify the topology of the network when some nodes are not contaminated by noise. One of the drawbacks of the methods mentioned before is the computational complexity. Therefore, new tools need to be developed for the identification of large networks of interconnected systems with general topology. In the same line, Torres et al. (2015) proposes an efficient Output-Error identification method for 1D interconnected systems. The method exploits Sequentially Semi-Separable (SSS) matrix operations to speed up the optimization routines. Moreover, in Yu and Verhaegen (2017) a subspace identification approach for 1D heterogeneous interconnected systems that exploit local input-output data is presented. Even though these methods are efficient in terms of computational complexity they focus in 1D spatially interconnected systems and not in complexer networks.

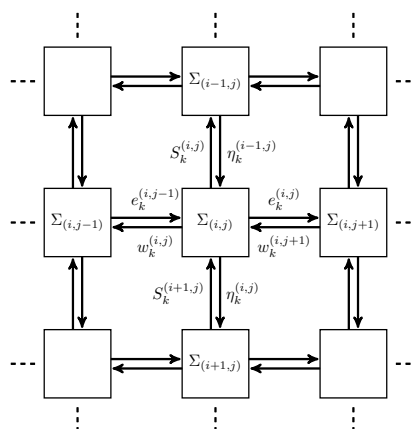


Fig. 1. 2D spatially interconnected systems.

In order to cope with this limitation, in this paper, an efficient Output-Error identification method for 2D grids of interconnected systems is introduced. The algorithm exploits the Multilevel Sequentially Semi-Separable (MSSS) structure, Qiu et al. (2015), of the matrices involved to perform an efficient Steepest-Descent optimization in linear complexity with respect to the number of subsystems in the grid. This is achieved by using basic operations with MSSS matrices. The main contribution of this paper is to show that all the matrices involved in the computations (including partial derivatives) have MSSS structure.

In this context, the outline of our paper is as follows: In Section 2, the problem formulation and the statement of the identification problem are presented. The complete algorithm is exposed in Section 3 while in Section 4 an example is provided in order to show the potential of the proposed algorithm. Finally, we draw our conclusions.

$$\begin{aligned}
 \text{a) } \overline{A}^{(1)} &= \begin{bmatrix} A_{(1,1)} & B_w^{(1,1)} C_w^{(1,2)} & B_w^{(1,1)} W_w^{(1,2)} C_w^{(1,3)} & B_w^{(1,1)} W_w^{(1,2)} W_w^{(1,3)} C_w^{(1,4)} \\ B_e^{(1,2)} C_e^{(1,1)} & A_{(1,2)} & B_w^{(1,2)} C_w^{(1,3)} & B_w^{(1,2)} W_w^{(1,3)} C_w^{(1,4)} \\ B_e^{(1,3)} W_e^{(1,2)} C_e^{(1,1)} & B_e^{(1,3)} C_e^{(1,2)} & A_{(1,3)} & B_w^{(1,3)} C_w^{(1,4)} \\ B_e^{(1,4)} W_e^{(1,3)} W_e^{(1,2)} C_e^{(1,1)} & B_e^{(1,4)} W_e^{(1,3)} C_e^{(2,1)} & B_e^{(1,4)} C_e^{(1,3)} & A_{(1,4)} \end{bmatrix} \\
 \text{b) } \overline{\overline{A}} &= \begin{bmatrix} A_{(1,1)} & B_w^{(1,1)} C_w^{(1,2)} & B_w^{(1,1)} W_w^{(1,2)} C_w^{(1,3)} & B_s^{(1,1)} C_s^{(2,1)} & 0 & 0 & B_s^{(1,1)} W_s^{(2,1)} C_s^{(3,1)} & 0 & 0 \\ B_e^{(1,2)} C_e^{(1,1)} & A_{(1,2)} & B_w^{(1,2)} C_w^{(1,3)} & 0 & B_s^{(1,2)} C_s^{(2,2)} & 0 & 0 & B_s^{(1,2)} W_s^{(2,2)} C_s^{(3,2)} & 0 \\ B_e^{(1,3)} W_e^{(1,2)} C_e^{(1,1)} & B_e^{(1,3)} C_e^{(1,2)} & A_{(1,3)} & 0 & 0 & B_s^{(1,3)} C_s^{(2,3)} & 0 & 0 & B_s^{(1,3)} W_s^{(2,3)} C_s^{(3,3)} \\ \\ B_N^{(2,1)} C_N^{(1,1)} & 0 & 0 & A_{(2,1)} & B_w^{(2,1)} C_w^{(2,2)} & B_w^{(2,1)} W_w^{(2,2)} C_w^{(2,3)} & B_s^{(2,1)} C_s^{(3,1)} & 0 & 0 \\ 0 & B_N^{(2,2)} C_N^{(1,2)} & 0 & B_e^{(2,2)} C_e^{(2,1)} & A_{(2,2)} & B_w^{(2,2)} C_w^{(2,3)} & 0 & B_s^{(2,2)} C_s^{(3,2)} & 0 \\ 0 & 0 & B_N^{(2,3)} C_N^{(1,3)} & B_e^{(2,3)} W_e^{(2,2)} C_e^{(2,1)} & B_e^{(2,3)} C_e^{(2,2)} & A_{(2,3)} & 0 & 0 & B_s^{(2,3)} C_s^{(3,3)} \\ \\ B_N^{(3,1)} W_N^{(2,1)} C_N^{(1,1)} & 0 & 0 & B_N^{(3,1)} C_N^{(2,1)} & 0 & 0 & A_{(3,1)} & B_w^{(3,1)} C_w^{(3,2)} & B_w^{(3,1)} W_w^{(3,2)} C_w^{(3,3)} \\ 0 & B_N^{(3,2)} W_N^{(2,2)} C_N^{(1,2)} & 0 & 0 & B_N^{(3,2)} C_N^{(2,2)} & 0 & B_e^{(3,2)} C_e^{(3,1)} & A_{(3,2)} & B_w^{(3,2)} C_w^{(3,3)} \\ 0 & 0 & B_N^{(3,3)} W_N^{(2,3)} C_N^{(1,3)} & 0 & 0 & B_N^{(3,3)} C_N^{(2,3)} & B_e^{(3,3)} W_e^{(3,2)} C_e^{(3,1)} & B_e^{(3,3)} C_e^{(3,2)} & A_{(3,3)} \end{bmatrix}
 \end{aligned}$$

Fig. 2. a) Sequentially Semi-Separable matrix $\overline{A}^{(1)}$ for $N = 4$, b) Multi-Level Sequentially Semi-Separable system matrix $\overline{\overline{A}}$ for $N = 3$ y $M = 3$

2. PROBLEM FORMULATION

We consider a 2D grid of $M \times N$ interconnected systems as depicted in Fig.1. Every sub-system is described by the following set of equations:

$$\Sigma_{(i,j)} : \begin{bmatrix} x_{k+1}^{(i,j)} \\ y_k^{(i,j)} \\ e_k^{(i,j)} \\ w_k^{(i,j)} \\ \eta_k^{(i,j)} \\ S_k^{(i,j)} \end{bmatrix} = \begin{bmatrix} A_{(i,j)} & B_{(i,j)} & B_e^{(i,j)} & B_w^{(i,j)} & B_N^{(i,j)} & B_s^{(i,j)} \\ C_{(i,j)} & D_{(i,j)} & F_e^{(i,j)} & F_w^{(i,j)} & F_N^{(i,j)} & F_s^{(i,j)} \\ C_e^{(i,j)} & D_e^{(i,j)} & W_e^{(i,j)} & 0 & 0 & 0 \\ C_w^{(i,j)} & D_w^{(i,j)} & 0 & W_w^{(i,j)} & 0 & 0 \\ C_N^{(i,j)} & D_N^{(i,j)} & 0 & 0 & W_N^{(i,j)} & 0 \\ C_s^{(i,j)} & D_s^{(i,j)} & 0 & 0 & 0 & W_s^{(i,j)} \end{bmatrix} \begin{bmatrix} x_k^{(i,j)} \\ u_k^{(i,j)} \\ e_{k-1}^{(i,j)} \\ w_k^{(i,j+1)} \\ \eta_{k-1}^{(i,j)} \\ S_k^{(i+1,j)} \end{bmatrix}, \quad (1)$$

where $x_k^{(i,j)} \in \mathbb{R}^n$ is the state vector, $u_k^{(i,j)} \in \mathbb{R}^r$ is the input vector, $y_k^{(i,j)} \in \mathbb{R}^\ell$ is the output vector, $e_k^{(i,j)} \in \mathbb{R}^q$, $w_k^{(i,j)} \in \mathbb{R}^q$, $\eta_k^{(i,j)} \in \mathbb{R}^q$ and $S_k^{(i,j)} \in \mathbb{R}^q$ are the interconnection variables¹. The dimensions of the matrices in (1) are consistent to the sizes of the corresponding vectors.

If we consider $N \times M$ sub-systems with the following boundary conditions: $e_k^{(0,j)} = 0$, $w_k^{(N+1,j)} = 0$, $\eta_k^{(i,0)} = 0$ and $S_k^{(i,M+1)} = 0$, ($\forall i = 1, 2, \dots, M$, $\forall j = 1, 2, \dots, N$) the lifted system representing the complete 2D grid can be obtained by performing consecutive substitutions in (1):

$$\overline{\Sigma} : \begin{bmatrix} \overline{x}_{k+1} \\ \overline{y}_k \end{bmatrix} = \begin{bmatrix} \overline{A} & \overline{B} \\ \overline{C} & \overline{D} \end{bmatrix} \begin{bmatrix} \overline{x}_k \\ \overline{u}_k \end{bmatrix}, \quad (2)$$

where $\overline{x}_k \in \mathbb{R}^{NMn}$ is the interconnected state vector that contains all the local states stacked together. Similar definitions hold for $\overline{x}_{k+1} \in \mathbb{R}^{NMn}$, $\overline{u}_k \in \mathbb{R}^{NM r}$ and $\overline{y}_k \in \mathbb{R}^{NM \ell}$. On the other hand, $\overline{A} \in \mathbb{R}^{NMn \times NMn}$, $\overline{B} \in \mathbb{R}^{NMn \times NM r}$, $\overline{C} \in \mathbb{R}^{NM \ell \times NMn}$, and $\overline{D} \in \mathbb{R}^{NM \ell \times NM r}$ are Multilevel Sequentially Semi-Separable (MSSS) matrices

¹ Without loss of generality, we assume the subsystems have the same number of states, inputs, outputs and interconnection variables.

characterized by the special structure depicted in Fig. 2.b). Notice from Fig. 2.b) that a MSSS matrix is a Sequentially Semi-Separable (SSS) matrix whose generators are SSS matrices themselves (see eq. (4)-(10)). Moreover, an SSS matrix has the structure depicted in Fig. 2.a). This matrix can be obtained by sequentially multiplying the generators appearing in eq. (4) as shown in Fig. 2.a) and by placing the matrices $A^{(i,j)}$ in the main diagonal.

$$\overline{\overline{A}} = MSSS(\overline{B}_N, \overline{W}_N, \overline{C}_N, \overline{A}^{(i)}, \overline{B}_s, \overline{W}_s, \overline{C}_s^{(i)}), \quad (3)$$

$$\overline{A}^{(i)} = SSS(B_e^{(i,j)}, W_e^{(i,j)}, C_e^{(i,j)}, A^{(i,j)}, B_w^{(i,j)}, W_w^{(i,j)}, C_w^{(i,j)}), \quad (4)$$

$$\overline{B}_N^{(i)} = SSS(0, 0, 0, B_N^{(i,j)}, 0, 0, 0), \quad (5)$$

$$\overline{W}_N^{(i)} = SSS(0, 0, 0, W_N^{(i,j)}, 0, 0, 0), \quad (6)$$

$$\overline{C}_N^{(i)} = SSS(0, 0, 0, C_N^{(i,j)}, 0, 0, 0), \quad (7)$$

$$\overline{B}_s^{(i)} = SSS(0, 0, 0, B_s^{(i,j)}, 0, 0, 0), \quad (8)$$

$$\overline{W}_s^{(i)} = SSS(0, 0, 0, W_s^{(i,j)}, 0, 0, 0), \quad (9)$$

$$\overline{C}_s^{(i)} = SSS(0, 0, 0, C_s^{(i,j)}, 0, 0, 0). \quad (10)$$

$$\forall i \in \{1, \dots, M\}, \forall j \in \{1, \dots, N\}$$

Therefore, for the rest of the lifted matrices we have a similar definition but exchanging the corresponding generators. Thus for obtaining \overline{B} we replace matrix A by B and C by D in eq. (3)-(10). For \overline{C} we replace matrix A by C and B by F and finally for \overline{D} we replace matrices A and C by D and B by F .

Thus, the identification problem can now be formulated as follows: Given the input-output sequences:

$$\{u_k^{(i,j)}, y_k^{(i,j)}\}_{k=\{1, \dots, K\}, i=\{1, \dots, M\}, j=\{1, \dots, N\}},$$

estimate the distributed system matrices $A_{(i,j)}$, $B_{(i,j)}$, $B_e^{(i,j)}$, $B_w^{(i,j)}$, $B_N^{(i,j)}$, $B_s^{(i,j)}$, $C_{(i,j)}$, $C_e^{(i,j)}$, $C_w^{(i,j)}$, $C_N^{(i,j)}$, $C_s^{(i,j)}$, $D_{(i,j)}$, $D_e^{(i,j)}$, $D_w^{(i,j)}$, $D_N^{(i,j)}$, $D_s^{(i,j)}$, $F_e^{(i,j)}$, $F_w^{(i,j)}$,

$F_N^{(i,j)}$, $F_s^{(i,j)}$, $W_e^{(i,j)}$, $W_w^{(i,j)}$, $W_N^{(i,j)}$ and $W_s^{(i,j)}$ for all $i \in \{1, 2, \dots, M\}$ and $j \in \{1, 2, \dots, N\}$ up to a set of similarity transformations.

3. DISTRIBUTED OUTPUT-ERROR PARAMETRIC MODEL ESTIMATION

In this section, the Multilevel Sequentially Semi-Separable (MSSS) Output-Error identification approach is presented. Firstly, the Steepest-Descent optimization method is introduced. Then, the use of MSSS matrices is described together with an analysis of computational complexity. It is shown that all the computations of the algorithm can be expressed as basic operations with MSSS matrices and thus they can be performed in linear complexity with respect to the number of subsystems in the grid.

3.1 Numerical optimization

In the Output-Error model approach, the measurements are supposed to be contaminated by white noise. This fact leads to the derivation of the following objective function:

$$J_K(\bar{\theta}) = \frac{1}{K} \sum_{k=1}^K \|\bar{y}_k - \hat{y}_k(\bar{\theta})\|_2^2 = \frac{1}{K} E_K^T(\bar{\theta}) E_K(\bar{\theta}), \quad (11)$$

where

$$E_K(\bar{\theta}) = \begin{bmatrix} \bar{y}_1 - \hat{y}_1(\bar{\theta}) \\ \bar{y}_2 - \hat{y}_2(\bar{\theta}) \\ \vdots \\ \bar{y}_K - \hat{y}_K(\bar{\theta}) \end{bmatrix} = \begin{bmatrix} \epsilon_1(\bar{\theta}) \\ \epsilon_2(\bar{\theta}) \\ \vdots \\ \epsilon_K(\bar{\theta}) \end{bmatrix},$$

\bar{y}_k is the k-th measurement of the output and $\hat{y}_k(\bar{\theta})$ is the output of the model. In order to minimize the objective function given in (11), several optimization methods can be considered (Steepest-Descent, Quasi-Newton), for the sake of simplicity, in this paper, we focus on the Steepest-Descent approach and therefore the computation of the Jacobian is required:

$$J'_K = \frac{2}{K} \left(\frac{\partial E_K(\bar{\theta})}{\partial \bar{\theta}} \right)^T E_K(\bar{\theta}) \quad (12)$$

For the state-space model considered in this paper, we use the following equation for the Jacobian which includes the adjoint state \bar{X}_k :

$$J'_K = -\frac{2}{K} [J'_1 + J'_2 + J'_3 + J'_4], \quad (13)$$

with:

$$J'_1 = \sum_{k=1}^K \left(\frac{\partial \bar{A}(\bar{\theta})}{\partial \bar{\theta}} \left[I_p \otimes \hat{x}_k(\bar{\theta}) \right] \right)^T \bar{X}_k(\bar{\theta}), \quad (14)$$

$$J'_2 = \sum_{k=1}^K \left(\frac{\partial \bar{B}(\bar{\theta})}{\partial \bar{\theta}} \left[I_p \otimes \bar{u}_k \right] \right)^T \bar{X}_k(\bar{\theta}), \quad (15)$$

$$J'_3 = \sum_{k=1}^K \left(\frac{\partial \bar{C}(\bar{\theta})}{\partial \bar{\theta}} \left[I_p \otimes \hat{x}_k(\bar{\theta}) \right] \right)^T \epsilon_k(\bar{\theta}), \quad (16)$$

$$J'_4 = \sum_{k=1}^K \left(\frac{\partial \bar{D}(\bar{\theta})}{\partial \bar{\theta}} \left[I_p \otimes \bar{u}_k \right] \right)^T \epsilon_k(\bar{\theta}), \quad (17)$$

and the adjoint state-space equation given by:

$$\bar{X}_{k-1}(\bar{\theta}) = \bar{A}^T(\bar{\theta}) \bar{X}_k(\bar{\theta}) + \bar{C}^T(\bar{\theta}) \epsilon_k(\bar{\theta}) \quad (18)$$

where $\hat{x}_k(\bar{\theta})$ and $\epsilon_k(\bar{\theta})$ are obtained by simulating the model given in (2). Notice that in this case two simulations are needed: the backward simulation (18) and the forward simulation of (2). For the derivation of the equations presented above the interested reader is referred to Verhaegen and Verdult (2007).

The main contribution of this paper is to show that all the matrices involved in the computations have MSSS structure (denoted by $\bar{\mathfrak{F}}$). Furthermore, the partial derivatives involved in the formulas can also be written as MSSS matrices:

$$\frac{\partial \bar{A}}{\partial \bar{\theta}} \left[I_p \otimes \hat{x}_k(\bar{\theta}) \right] = MSSS(\bar{B}_N^{(i)}, \bar{W}_N^{(i)}, \bar{C}_N^{(i)}, \bar{A}^{(i)}, \bar{B}_s^{(i)}, \bar{W}_s^{(i)} \bar{C}_s^{(i)}),$$

$$\frac{\partial \bar{B}}{\partial \bar{\theta}} \left[I_p \otimes \bar{u}_k \right] = MSSS(\bar{B}_N^{(i)}, \bar{W}_N^{(i)}, \bar{D}_N^{(i)}, \bar{B}^{(i)}, \bar{B}_s^{(i)}, \bar{W}_s^{(i)} \bar{D}_s^{(i)}),$$

$$\frac{\partial \bar{C}}{\partial \bar{\theta}} \left[I_p \otimes \hat{x}_k(\bar{\theta}) \right] = MSSS(\bar{F}_N^{(i)}, \bar{W}_N^{(i)}, \bar{C}_N^{(i)}, \bar{C}^{(i)}, \bar{F}_s^{(i)}, \bar{W}_s^{(i)} \bar{C}_s^{(i)}),$$

$$\frac{\partial \bar{D}}{\partial \bar{\theta}} \left[I_p \otimes \bar{u}_k \right] = MSSS(\bar{F}_N^{(i)}, \bar{W}_N^{(i)}, \bar{D}_N^{(i)}, \bar{D}^{(i)}, \bar{F}_s^{(i)}, \bar{W}_s^{(i)} \bar{D}_s^{(i)}),$$

$$\forall i \in \{1, \dots, M\}$$

$$\bar{A}^{(i)} = SSS(B_e^{(i,j)}, W_e^{(i,j)}, C_e^{(i,j)}, \mathcal{A}_{(i,j)}, B_w^{(i,j)}, W_w^{(i,j)}, C_w^{(i,j)}),$$

$$\bar{B}^{(i)} = SSS(B_e^{(i,j)}, W_e^{(i,j)}, \mathcal{D}_{(i,j)}, \mathcal{B}_{(i,j)}, B_w^{(i,j)}, W_w^{(i,j)}, \mathcal{D}_w^{(i,j)}),$$

$$\bar{C}^{(i)} = SSS(F_e^{(i,j)}, W_e^{(i,j)}, C_e^{(i,j)}, \mathcal{C}_{(i,j)}, F_w^{(i,j)}, W_w^{(i,j)}, C_w^{(i,j)}),$$

$$\bar{D}^{(i)} = SSS(F_e^{(i,j)}, W_e^{(i,j)}, \mathcal{D}_{(i,j)}, \mathcal{D}_{(i,j)}, F_w^{(i,j)}, W_w^{(i,j)}, \mathcal{D}_w^{(i,j)}),$$

$$\bar{C}_N^{(i)} = SSS(0, 0, 0, C_N^{(i,j)}, 0, 0, 0),$$

$$\bar{C}_s^{(i)} = SSS(0, 0, 0, C_s^{(i,j)}, 0, 0, 0),$$

$$\bar{D}_N^{(i)} = SSS(0, 0, 0, \mathcal{D}_N^{(i,j)}, 0, 0, 0),$$

$$\bar{D}_s^{(i)} = SSS(0, 0, 0, \mathcal{D}_s^{(i,j)}, 0, 0, 0),$$

$$\forall j \in \{1, \dots, N\}$$

where the generator matrices in caligraphic style are depicted in Fig. 3. The corresponding matrix recurrences γ and δ used to compute the generators are given in Appendix A.

3.2 Computational Complexity

According to the exposed before, matrix-vector product and matrix transpose MSSS operations are required for the two simulations and the computation of the Jacobian in every iteration. Furthermore, both operations can be performed in linear complexity with respect to the number of subsystems in the grid $\mathcal{O}(NM)$, (Houtzager and Rice (2009)). Moreover, the identification problem described above requires $p = (n^2 + nr + 8nq + \ell n + \ell r + 4lq + 4rq + 4q^2)NM$ parameters to be estimated. This is clearly much better than the complexity of the full parameterization of the state space model (2), that produces $(n^2 + nr + n\ell + \ell r)N^2M^2$ parameters (assuming that $n, r, \ell, q \ll N, M$). In addition, the number of parameters to be stored for the partial derivatives also increase linearly with respect to the number of sub-systems in the grid: $(2n + 2\ell + 12q)(n^2 + nr + 8nq + n\ell + \ell r + 4qr + 4ql + 4q^2)NM$ (see Fig. 3). In the case of Quasi-Newton methods, the implementation also

$$\begin{aligned}
 & \left[\mathcal{A}_{(i,j)}^T \mid \mathcal{B}_{(i,j)}^T \mid \mathcal{C}_{(i,j)}^T \mid \mathcal{D}_{(i,j)}^T \mid \mathcal{C}_e^{(i,j)T} \mid \mathcal{C}_w^{(i,j)T} \mid \mathcal{C}_N^{(i,j)T} \mid \mathcal{C}_s^{(i,j)T} \mid \mathcal{D}_e^{(i,j)T} \mid \mathcal{D}_w^{(i,j)T} \mid \mathcal{D}_N^{(i,j)T} \mid \mathcal{D}_s^{(i,j)T} \right]^T = [G_1 \mid G_2] \\
 & G_1 = \begin{bmatrix} x^{(i,j)T} \otimes I_n & 0_{n \times nr} & \gamma_e^{(i,j)} \otimes I_n & \gamma_w^{(i,j)} \otimes I_n & \gamma_N^{(i,j)} \otimes I_n & \gamma_s^{(i,j)} \otimes I_n & 0_{n \times nl} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times lr} \\ 0_{n \times n^2} & u^{(i,j)T} \otimes I_n & \delta_e^{(i,j)} \otimes I_n & \delta_w^{(i,j)} \otimes I_n & \delta_N^{(i,j)} \otimes I_n & \delta_s^{(i,j)} \otimes I_n & 0_{n \times nl} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times nq} & 0_{n \times lr} \\ 0_{\ell \times n^2} & 0_{\ell \times nr} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & x^{(i,j)T} \otimes I_\ell & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times lr} \\ 0_{\ell \times n^2} & 0_{\ell \times nr} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nl} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & 0_{\ell \times nq} & u^{(i,j)T} \otimes I_\ell \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & x^{(i,j)T} \otimes I_q & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & x^{(i,j)T} \otimes I_q & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & x^{(i,j)T} \otimes I_q & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & x^{(i,j)T} \otimes I_q & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \\ 0_{q \times n^2} & 0_{q \times nr} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nl} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times nq} & 0_{q \times lr} \end{bmatrix} \\
 & G_2 = \begin{bmatrix} 0_{n \times qr} & 0_{n \times qr} & 0_{n \times qr} & 0_{n \times qr} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times q^2} & 0_{n \times q^2} & 0_{n \times q^2} & 0_{n \times q^2} \\ 0_{n \times qr} & 0_{n \times qr} & 0_{n \times qr} & 0_{n \times qr} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times ql} & 0_{n \times q^2} & 0_{n \times q^2} & 0_{n \times q^2} & 0_{n \times q^2} \\ 0_{\ell \times qr} & 0_{\ell \times qr} & 0_{\ell \times qr} & 0_{\ell \times qr} & \gamma_e^{(i,j)} \otimes I_\ell & \gamma_w^{(i,j)} \otimes I_\ell & \gamma_N^{(i,j)} \otimes I_\ell & \gamma_s^{(i,j)} \otimes I_\ell & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} \\ 0_{\ell \times qr} & 0_{\ell \times qr} & 0_{\ell \times qr} & 0_{\ell \times qr} & \delta_e^{(i,j)} \otimes I_\ell & \delta_w^{(i,j)} \otimes I_\ell & \delta_N^{(i,j)} \otimes I_\ell & \delta_s^{(i,j)} \otimes I_\ell & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} & 0_{\ell \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & \gamma_e^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & \gamma_w^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & 0_{q \times q^2} & \gamma_N^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} & \gamma_s^{(i,j)} \otimes I_q & 0_{q \times q^2} \\ u^{(i,j)T} \otimes I_q & 0_{q \times nr} & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & \delta_e^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & u^{(i,j)T} \otimes I_q & 0_{q \times qr} & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & \delta_w^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & u^{(i,j)T} \otimes I_q & 0_{q \times qr} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & 0_{q \times q^2} & \delta_N^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} \\ 0_{q \times qr} & 0_{q \times qr} & 0_{q \times qr} & u^{(i,j)T} \otimes I_q & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times ql} & 0_{q \times q^2} & 0_{q \times q^2} & \delta_s^{(i,j)} \otimes I_q & 0_{q \times q^2} & 0_{q \times q^2} \end{bmatrix}
 \end{aligned}$$

 Fig. 3. Generator matrices used to compute the partial derivatives (\otimes is the Kronecker product)

results in linear complexity with respect to the number of subsystems in the grid as the approximation of the Hessian can also be represented as a SSS matrix. In addition, faster convergence to the optimum should be expected.

4. SIMULATION EXAMPLE

The proposed identification algorithm is tested on the heat equation discretized in a 2D domain with Dirichlet boundary conditions:

$$\begin{bmatrix} x_{k+1}^{(i,j)} \\ y_k^{(i,j)} \\ e_k^{(i,j)} \\ w_k^{(i,j)} \\ \eta_k^{(i,j)} \\ S_k^{(i,j)} \end{bmatrix} = \begin{bmatrix} a_{(i,j)} & b_{(i,j)} & c_{(i,j)} & d_{(i,j)} & e_{(i,j)} & f_{(i,j)} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^{(i,j)} \\ u_k^{(i,j)} \\ e_{k-1}^{(i,j)} \\ w_{k-1}^{(i,j)} \\ \eta_{k-1}^{(i,j)} \\ S_{k-1}^{(i,j)} \end{bmatrix}, \quad (19)$$

$$c_{(i,j)} = 0.006(\Gamma_{(i,j)} + \Gamma_{(i,j-1)}), \quad (20)$$

$$d_{(i,j)} = 0.006(\Gamma_{(i,j)} + \Gamma_{(i,j+1)}), \quad (21)$$

$$e_{(i,j)} = 0.006(\Gamma_{(i,j)} + \Gamma_{(i-1,j)}), \quad (22)$$

$$f_{(i,j)} = 0.006(\Gamma_{(i,j)} + \Gamma_{(i+1,j)}) \quad (23)$$

where $a_{(i,j)} = 1 - (c_{(i,j)} + d_{(i,j)} - (e_{(i,j)} + f_{(i,j)})) - 0.5$, $b_{(i,j)} = 0.3 \times 1e - 06$ and $\Gamma_{(i,j)} \sim \mathcal{U}(0, 5)$ are uniformly distributed parameters. The input signals are normally distributed random variables with covariance matrix equal to $cov(u_k) = I_r$. In addition, a Gaussian noise is added to the outputs with a prescribed Signal-to-Noise Ratio (SNR).

In the following subsections, the prediction capabilities of the identified model, the computational complexity of the

algorithm and the effect of the noise level on the prediction error are investigated.

4.1 Variance Accounted For as function of the number of iterations

In this case, the Variance-Accounted-For (VAF)² is studied as function of the number of iterations. A 2D grid of $15 \times 15 = 225$ sub-systems is considered. A set of 20 experiments is performed where every trial is a random realization of the parameters of every sub-system in the grid. In addition, the outputs are corrupted with a Gaussian noise of $SNR_{dB} = 40dB$. The mean of the VAF is calculated over 20 trials and depicted in Fig. A.1. It is observed that the VAF converges to the 100% as the number of iterations increases.

4.2 Computational Complexity

In this section, the computation time per iteration of the proposed algorithm is studied. The computation time is recorded for 2D grids of increasing number of sub-systems and for 20 Monte-Carlo simulations per each grid. The results are presented in Fig. A.2.

In fact, it is observed that the computation time increases linearly as the number of sub-systems grows. This result confirms our hypothesis that the computations are performed on linear time with respect to the number of sub-systems in the grid $\mathcal{O}(NM)$.

² the VAF is defined as: $VAF = 100 \cdot \max\{0, (1 - \frac{var(\hat{y}_k - y_k)}{var(y_k)})\}[\%]$

4.3 The effect of noise

Finally, the effect of noise is investigated on the VAF index. The outputs are contaminated with Gaussian noise of different SNR characteristics and with different number of samples. Moreover, different realization of the subsystems in the grid along 20 Monte-Carlo simulations are considered. The results for two different data sets are depicted in Fig. A.3: A) noisy data used for training the model and B) noise-free data used for validation.

In this scenario, two effects can be observed. If the number of samples remain constant at $K = 30$, the VAF diminishes when the power of the noise increases (SNR decreases). This means that the prediction performance is deteriorated when the influence of noise is predominant. On the other hand, if the noise power is set constant and high at $\text{SNR}_{dB} = 10\text{dB}$, the prediction capabilities are enhanced (VAF is increased) if the amount of samples used on the identification is increased ($K = 90$), i.e, the use of longer sequences in the training procedure attenuates the noise overfit improving the results on the validation data.

5. CONCLUSION

In this paper, an efficient Output-Error identification method for 2D interconnected systems based on the Steepest-Descent gradient method was presented. It was shown that for 2D interconnected systems the whole algorithm can be expressed within the MSSS-approach which resulted in linear time computations with respect to the number of sub-systems in the grid. In addition, the results were strengthened by a simulation example showing the linearity and effectiveness of the proposed method. Convergence of the VAF to the 100% in finite number of iterations was exposed as well. Even though the prediction capabilities of the resulting model were deteriorated by output data contaminated with increasing noise power (overfit), this negative effects can be attenuated by considering data sequences with larger number of samples.

Appendix A

A.1 Matrices γ

Given the matrices $W_N \in \mathbb{R}^{q \times q}$ and $C_N \in \mathbb{R}^{q \times n}$ we define the matrix sequence $\gamma_N^{(i,j)}$ as follows:

$$\gamma_N^{(i,j)} = \gamma_N^{(i-1,j)} W_N^{(i-1,j)T} + x_{(i-1,j)}^T C_N^{(i-1,j)T},$$

$$\forall i \in \{2, \dots, M\}, \forall j \in \{1, \dots, N\}$$

$$\gamma_N^{(1,j)} = 0, \quad \forall j \in \{1, \dots, N\}$$

On the other hand, given $W_s \in \mathbb{R}^{q \times q}$ and $C_s \in \mathbb{R}^{q \times n}$ we have:

$$\gamma_s^{(i-1,j)} = \gamma_s^{(i,j)} W_s^{(i,j)T} + x_{(i,j)}^T C_s^{(i,j)T},$$

$$\forall i \in \{M, \dots, 2\}, \forall j \in \{1, \dots, N\}$$

$$\gamma_s^{(M,j)} = 0, \quad \forall j \in \{1, \dots, N\}$$

Finally, given $W_e \in \mathbb{R}^{q \times q}$, $W_w \in \mathbb{R}^{q \times q}$, $C_e \in \mathbb{R}^{q \times n}$ and $C_w \in \mathbb{R}^{q \times n}$ we define:

$$\gamma_e^{(i,j)} = \gamma_e^{(i,j-1)} W_e^{(i,j-1)T} + x_{(i,j-1)}^T C_e^{(i,j-1)T},$$

$$\forall i \in \{1, \dots, M\}, \forall j \in \{2, \dots, N\}$$

$$\gamma_e^{(i,1)} = 0, \quad \forall i \in \{1, \dots, M\}$$

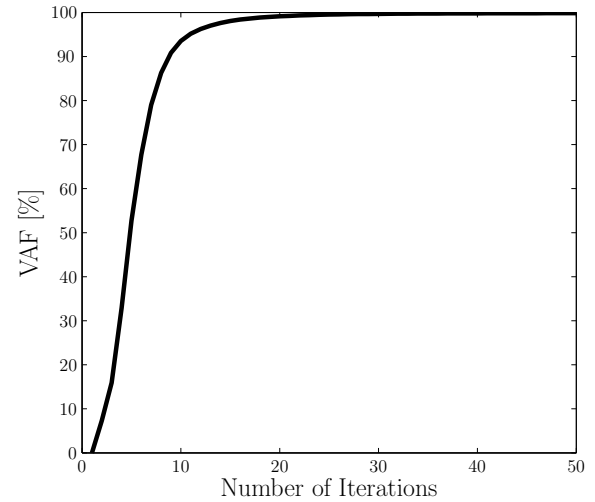


Fig. A.1. VAF Convergence.

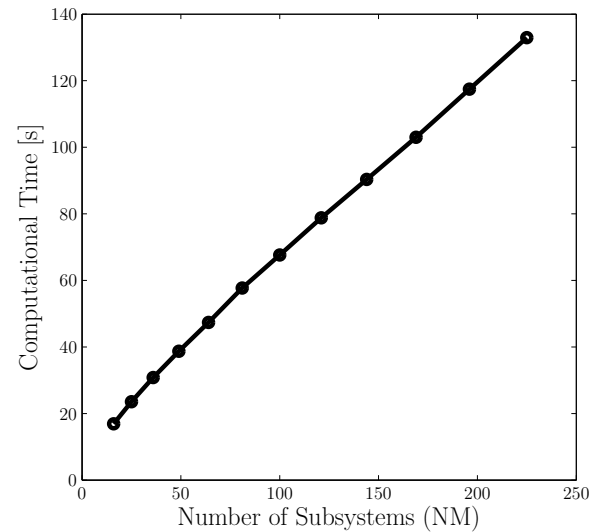


Fig. A.2. Computation time.

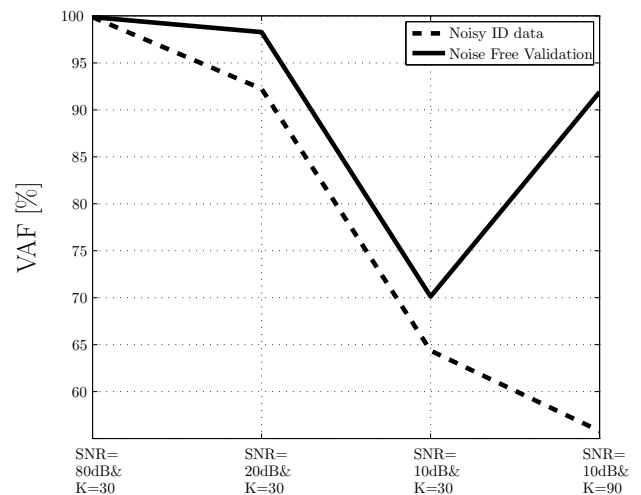


Fig. A.3. Effect of the noise.

and:

$$\begin{aligned}\gamma_w^{(i,j-1)} &= \gamma_w^{(i,j)} W_w^{(i,j)T} + x_{(i,j)}^T C_w^{(i,j)T}, \\ &\quad \forall i \in \{1, \dots, M\}, \forall j \in \{N, \dots, 2\} \\ \gamma_w^{(i,N)} &= 0. \quad \forall i \in \{1, \dots, M\}\end{aligned}$$

A.2 Matrices δ

Moreover, given the matrices $W_N \in \mathbb{R}^{q \times q}$ and $D_N \in \mathbb{R}^{q \times r}$ we define the matrix sequence $\delta_N^{(i,j)}$ as follows:

$$\begin{aligned}\delta_N^{(i,j)} &= \delta_N^{(i-1,j)} W_N^{(i-1,j)T} + u_{(i-1,j)}^T D_N^{(i-1,j)T}, \\ &\quad \forall i \in \{2, \dots, M\}, \forall j \in \{1, \dots, N\} \\ \delta_N^{(1,j)} &= 0. \quad \forall j \in \{1, \dots, N\}\end{aligned}$$

On the other hand, given $W_s \in \mathbb{R}^{q \times q}$ and $D_s \in \mathbb{R}^{q \times r}$ we have:

$$\begin{aligned}\delta_s^{(i-1,j)} &= \delta_s^{(i,j)} W_s^{(i,j)T} + u_{(i,j)}^T D_s^{(i,j)T}, \\ &\quad \forall i \in \{M, \dots, 2\}, \forall j \in \{1, \dots, N\} \\ \delta_s^{(M,j)} &= 0. \quad \forall j \in \{1, \dots, N\}\end{aligned}$$

Finally, given $W_e \in \mathbb{R}^{q \times q}$, $W_w \in \mathbb{R}^{q \times q}$, $D_e \in \mathbb{R}^{q \times r}$ and $D_w \in \mathbb{R}^{q \times r}$ we define:

$$\begin{aligned}\delta_e^{(i,j)} &= \delta_e^{(i,j-1)} W_e^{(i,j-1)T} + u_{(i,j-1)}^T D_e^{(i,j-1)T}, \\ &\quad \forall i \in \{1, \dots, M\}, \forall j \in \{2, \dots, N\} \\ \delta_e^{(i,1)} &= 0, \quad \forall i \in \{1, \dots, M\}\end{aligned}$$

and:

$$\begin{aligned}\delta_w^{(i,j-1)} &= \delta_w^{(i,j)} W_w^{(i,j)T} + u_{(i,j)}^T D_w^{(i,j)T}, \\ &\quad \forall i \in \{1, \dots, M\}, \forall j \in \{N, \dots, 2\} \\ \delta_w^{(i,N)} &= 0. \quad \forall i \in \{1, \dots, M\}\end{aligned}$$

REFERENCES

- Cantoni, M., Weyer, E., Li, Y., Ooi, S., Mareels, I., and Ryan, M. (2007). Control of large scale irrigation networks. *Proceedings of IEEE*, 95(1), 75–91.
- Dankers, A. (2015). *System identification in dynamic networks*. Ph.D. thesis, Delft University of Technology.
- Fraanje, R., Rice, J., Verhaegen, M., and Doelman, N. (2010). Fast reconstruction and prediction of frozen flow turbulence based on structural Kalman filtering. *Journal of the Optical Society of America*, 27(11), 235–245.
- Horowitz, R. and Varaiya, P. (2000). Control design of an automated highway system. *Proceedings of IEEE*, 88(7), 913–925.
- Houtzager, I. and Rice, J. (2009). Multilevel sequentially semi-separable matrix toolbox for MATLAB.
- Massioni, P. (2010). *Decomposition methods for distributed control and identification*. Ph.D. thesis, Delft University of Technology.
- Mukhtar, A., Ahsan, A., Hossam, A., and Werner, H. (2011). Identification of Box-Jenkins models for parameter-varying spatially interconnected systems. In *Proceedings of the IEEE American Control Conference (ACC)*. San Francisco, CA, USA.
- Mukhtar, A., Saulat, C., and Werner, H. (2010). Identification of LPV models for spatially varying interconnected systems. In *Proceedings of the IEEE American Control Conference (ACC)*. Baltimore, MD, USA.

- Qiu, Y., Van Gijzen, M.B., Van Wingerden, J.W., Verhaegen, M., and Vuik, C. (2015). Efficient preconditioners for PDE-constrained optimization problem with a multilevel sequentially semiseparable matrix structure. *Electronic Transactions on Numerical Analysis*, 44, 367–400.
- Scholte, E. and D’Andrea, R. (2003). Active vibro-acoustic control of a flexible beam using distributed control. In *Proceedings of the IEEE American Control Conference (ACC)*. Denver, CO, USA.
- Torres, P., Van Wingerden, J.W., and Verhaegen, M. (2015). Output-error identification of large scale 1-d spatially varying interconnected systems. *IEEE Transactions on Automatic Control*, 60(1), 130–142.
- Verhaegen, M. and Verdult, V. (2007). *Filtering and System Identification: A least squares approach*. Cambridge University Press.
- Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2016). Identifiability of dynamic networks with part of the nodes noise-free. *IFAC-PapersOnLine*, 49(13), 19–24.
- Yu, C. and Verhaegen, M. (2017). Subspace identification of 1d large-scale heterogeneous network. In *Proceedings of the 13th IEEE Conference on Control and Automation (ICCA)*. Ohrid, Macedonia.