Kronecker-ARX models in identifying (2D) spatial-temporal systems

Sinquin, B.; Verhaegen, M.

DOI
10.1016/j.ifacol.2017.08.1855

Publication date
2017

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 14131–14136

Kronecker-ARX models in identifying (2D) spatial-temporal systems ⋆

B. Sinquin * **M. Verhaegen** *

* *Delft Center for Systems and Control*
Delft University of Technology
Delft, Netherlands.

Abstract: In this paper we address the identification of (2D) spatial-temporal dynamical systems governed by the Vector Auto-Regressive (VAR) form. The coefficient-matrices of the VAR model are parametrized as sums of Kronecker products. When the number of terms in the sum is small compared to the size of the matrix, such a Kronecker representation leads to high data compression. Estimating in least-squares sense the coefficient-matrices gives rise to a bilinear estimation problem, which is tackled using a three-stage algorithm. A numerical example demonstrates the advantages of the new modeling paradigm. It leads to comparable performances with the unstructured least-squares estimation of VAR models. However, the number of parameters in the new modeling paradigm grows linearly w.r.t. the number of nodes in the 2D sensor network instead of quadratically in the full unstructured matrix case.

Keywords: 2D large-scale systems, Vector AutoRegressive model, Kronecker product, low-rank approximation.

1. INTRODUCTION

Modeling large-scale networks has been stirring much developments in various fields such as machine learning and system identification. Networks with unknown interconnection pattern arise for example in fields such as biology, e.g with the modeling of brain neurons, or in optics while modeling the atmospheric turbulence. Due to the large size of input-output data batches, identifying locally the behavior of the network is a major challenge that has been mainly addressed by using prior knowledge on how the nodes, or subsystems, are connected to one another. One common assumption is sparsity and relies on the fact that each node is connected to a limited number of other nodes with respect to the network's size. In the so-called sparse plus low rank networks -Zorzi and Chiuso (2016)-, a few latent variables relate most of the measured nodes from which few of them influence each other. Model identification remains however computationally challenging to handle the combination of these two matrix structure. Other well-studied structures include interconnected one-dimensional strings of subsystems in Rice (2010), or clusters of different subsystems with known connection patterns, named as alpha-heterogeneous in Massioni (2014). However the links between the subsystems in the network might not be known beforehand.

Inspired by the multi-level structured matrices that arise in image processing, e.g deblurring, we introduce a new class of structured interconnected systems to model 2D spatial-temporal systems. Denoting an object O imaged with a static optical system, the resulting blurred image B undergoes the linear blurring operation as follows, Hansen (2006):

$$\text{vec}(B) = \mathbf{A}\text{vec}(O) \tag{1}$$

The coefficient-matrix $\mathbf{A}$ is related to the Point-Spread Function (PSF) (or 2D impulse response) of the optical system. The equation (1) represents the 2D convolution operation between the PSF and the object O. The structure in $\mathbf{A}$ is related to the separability of the PSF. For example, the PSF for blurring caused by atmospheric turbulence can be described as a two-dimensional Gaussian function, Roggemann and Welsh (1996). The latter function is separable in both horizontal and vertical coordinates, which implies the following Kronecker structure for the coefficient-matrix $\mathbf{A}$:

$$\mathbf{A} = A_r \otimes A_c \tag{2}$$

where A_r and A_c represent respectively the 1D convolution with the rows and columns. The Kronecker representation enables a data-sparse representation of $\mathbf{A}$ and fast computations thanks to the very pleasant algebra of the Kronecker product, van Loan (2000). For example with $A_r, A_c \in \mathbb{R}^{N\times N}$, matrix-matrix multiplication and inversion both require $\mathcal{O}(N^4)$ complexity instead of $\mathcal{O}(N^6)$ for the unstructured case. A large-scale static input-output map in (1) is represented by a Kronecker matrix as in (2). It provides with a motivation to analyze Kronecker networks with a dynamical systems perspective.

In this paper we address the identification of (2D) spatial-temporal dynamical models of the Vector Auto-Regressive (VAR) form. The coefficient-matrices $\{A_i\}_{i=1..p} \in \mathbb{R}^{N^2\times N^2}$ of this temporal model with order p are parametrized as sums of Kronecker products:

⋆ Corresponding author: b.sinquin@tudelft.nl.
The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 339681.

10.1016/j.ifacol.2017.08.1855

$$A_i = \sum_{i=1}^{r} U_i \otimes V_i$$

where $r \ll N^2$ and $U_i, V_i \in \mathbb{R}^{N\times N}$. The integer r is called the spatial order, or Kronecker rank. The matrices U_i, V_i are called the factor matrices.

For a given labeling of the 2D network, the coefficient-matrices in the VAR model may be represented with as few terms as possible in the Kronecker sum while still guaranteeing a given level of performance with respect to the standard least squares solution with unstructured coefficient matrices. A major challenge in the estimation of the Kronecker product matrices is the computational efficiency. Rather than estimating the full coefficient-matrices, the optimization is formulated on the factor matrices which gives rise to a bilinear least squares problem. We formulate here a three-stage algorithm whose non-regularized version is non-iterative. Such a formulation shares similarities with the identification of Hammerstein systems for which a two-stage algorithm was proposed in Wang (2009). A numerical example on Multi-Level Sequentially Semi-Separable (MSSS) matrices illustrates the performances of the Kronecker VAR identification.

This paper is organized as follows. In the second section the class of *Kronecker networks* is defined. The third section formulates the Kronecker VAR identification framework while the fourth section describes the non-iterative three-stage algorithm to estimate the factor matrices with minimum computational complexity. Section V is dedicated to a numerical example.

Notations. The vectorization operator for a matrix X is $\mathrm{vec}(X)$, shortly denoted in bold $\mathbf{x}$. $\mathrm{ivec}(\mathbf{x})$ reshapes the vector $\mathbf{x}$ into a matrix whose size will be clear from the context. The Kronecker product between two matrices X, Y is denoted by $X \otimes Y$. The 2-norm of a vector $\mathbf{x}$ is written as $\|\mathbf{x}\|_2$. The nuclear norm of X, denoted with $\|X\|_\star$, represents the sum of the singular values of X.

2. PRELIMINARIES

Definition 1. (van Loan (2000)). *Let X be a $m_1 \times n_1$ block matrix with blocks $X(i,j)$ in $\mathbb{R}^{m_2\times n_2}$, given as:*

$$X = \begin{bmatrix} X(1,1) & \cdots & X(1,n_1) \\ \vdots & \ddots & \vdots \\ X(m_1,1) & \cdots & X(m_1,n_1) \end{bmatrix} \in \mathbb{R}^{m_1m_2\times n_1n_2}$$

then the re-shuffle operator $\mathcal{R}(X)$ is defined as:

$$\mathcal{R}(X) = \begin{bmatrix} \mathrm{vec}\Big(X(1,1)\Big)^T \\ \vdots \\ \mathrm{vec}\Big(X(m_1,1)\Big)^T \\ \vdots \\ \mathrm{vec}\Big(X(1,n_1)\Big)^T \\ \vdots \\ \mathrm{vec}\Big(X(m_1,n_1)\Big)^T \end{bmatrix} \in \mathbb{R}^{m_1n_1\times m_2n_2}$$

Lemma 1. (van Loan (2000)). *Let X be defined as in Definition 1, and let $X = F \otimes G$, with $F, G \in \mathbb{R}^{m_1\times n_1} \times \mathbb{R}^{m_2\times n_2}$. Then:*

$$\mathcal{R}(X) = \mathrm{vec}(F)\mathrm{vec}(G)^T$$

The operation in Lemma 1 can also be reversed by the definition of the inverse vec operator, ivec(.).

Lemma 2. (van Loan (2000)). *Let X be defined as in Definition 1, and let a Singular Value Decomposition of $\mathcal{R}(X)$ be given as:*

$$\mathcal{R}(X) = \sum_{\ell=1}^{r} \sigma_\ell u_\ell v_\ell^T$$

and let $\mathrm{ivec}\Big(u_\ell\Big) = U_\ell$, $\mathrm{ivec}\Big(v_\ell\Big) = V_\ell$*:*

$$X = \sum_{\ell=1}^{r} \sigma_\ell U_\ell \otimes V_\ell$$

The integer r is called the *Kronecker rank* of X w.r.t. the chosen block partitioning of X as given in Definition 1.

Definition 2. (α-decomposable matrices, Massioni (2014)). *Let P be a $N \times N$ pattern matrix. Define $\beta_j = \sum_{i=1}^{j} N_i$ (with $\beta_0 = 0$) and $I_{[a_1:a_2]}$ as an $N \times N$ diagonal matrix which contains 1 in the diagonal entries of indices from a_1 to a_2 (included) and 0 elsewhere, then an α-decomposable matrix (for a given α) is a matrix of the following kind:*

$$\mathcal{M} = \sum_{i=1}^{\alpha} \Big(I_{[\beta_{i-1}+1:\beta_i]} \otimes M_a^{(i)} + I_{[\beta_{i-1}+1:\beta_i]} P \otimes M_b^{(i)} \Big)$$

The matrices $M_a^{(i)}$ are the diagonal blocks of $\mathcal{M}$, while the matrices $M_b^{(i)}$ constitute the off-diagonal blocks, according to the structure of P.

For $\alpha = 1$ (and $\beta_1 = N$), these matrices are simply called *decomposable* matrices. The class of α-decomposable matrices will be denoted by $\mathcal{D}^\alpha$, with for $\alpha = 1$ just the symbol $\mathcal{D}$ will be used. As a generalization of this class $\mathcal{D}^\alpha$ of structured matrices, we define next the class of sums of Kronecker product matrices.

Definition 3. The class of sums of Kronecker product matrices, denoted by $\mathcal{K}$, contains matrices of the following kind:

$$\mathcal{M} = \sum_{i=1}^{r} M_a^{(i)} \otimes M_b^{(i)}$$

with $M_a^{(i)} \in \mathbb{R}^{m_1\times n_1}$ and $M_b^{(i)} \in \mathbb{R}^{m_2\times n_2}$, and the Kronecker rank r such that $r \leq min(m_1n_1, m_2n_2)$.

Without further specification on the factor matrices, the network may be composed of heterogeneous subsystems.

3. PROBLEM FORMULATION

The sensor readings of a square N by N sensor grid at time instance k are stored in the matrix S_k as:

$$S_k = \begin{bmatrix} s_{1,1}(k) & s_{1,2}(k) & \cdots & s_{1,N}(k) \\ s_{2,1}(k) & s_{2,2}(k) & & s_{2,N}(k) \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1}(k) & s_{N,2}(k) & \cdots & s_{N,N}(k) \end{bmatrix}$$

In this paper we will consider that the temporal dynamics of this array of sensors is governed by the following VAR(X) model:

$$\mathrm{vec}\Big(S_k\Big) = \sum_{i=1}^{p} A_i \mathrm{vec}\Big(S_{k-i}\Big) + C_0 \mathrm{vec}\Big(E_k\Big) \tag{3}$$

with $\mathrm{vec}\Big(E_k\Big)$ zero-mean white noise with covariance matrix I. The coefficient matrices A_i and C_0 in the VARX (we will restrict for simplicity to the VAR-case) model (3), in general are highly structured. We will consider the case they belong to the set $\mathcal{K}$, and for the moment only focus on the coefficient matrices A_i. To address an identification problem we will parametrize these coefficient matrices as:

$$A_i = \sum_{j=1}^{r_i} M(b_i^{(j)})^T \otimes M(a_i^{(j)}) \tag{4}$$

with the vectors $a_i^{(j)}$ and $b_i^{(j)}$ parametrizing the matrices $M(a_i^{(j)})$ and $M(b_i^{(j)})$ in an affine manner. If we consider the term $C_0\mathbf{e}_k$ as a temporally white sequence $\mathbf{v}_k$, then the VARX model (3) can be written as,

$$\mathbf{s}_k = \sum_{i=1}^{p} \Big(\sum_{j=1}^{r_i} M(b_i^{(j)})^T \otimes M(a_i^{(j)}) \Big) \mathbf{s}_{k-i} + \mathbf{v}_k \tag{5}$$

Using the following Kronecker rule, for matrices X, Y, Z of compatible dimensions such that the product XYZ exists,

$$\mathrm{vec}\Big(XYZ\Big) = \Big(Z^T \otimes X\Big)\mathrm{vec}\Big(Y\Big)$$

we can write the ARX model (5) as,

$$S_k = \sum_{i=1}^{p} \Big(\sum_{j=1}^{r_i} M(a_i^{(j)}) S_{k-i} M(b_i^{(j)}) \Big) + V_k \tag{6}$$

This can also be written explicitly as,

$$S_k = \sum_{i=1}^{p} \mathcal{M}_{a,i} \Big(I_{r_i} \otimes S_{k-i} \Big) \mathcal{M}_{b,i} + V_k \tag{7}$$

where:

$$\mathcal{M}_{a,i} = \Big[M(a_i^{(1)}) \ \cdots \ M(a_i^{(r_i)}) \Big], \qquad \mathcal{M}_{b,i} = \begin{bmatrix} M(b_i^{(1)}) \\ \vdots \\ M(b_i^{(r_i)}) \end{bmatrix} \tag{8}$$

The AR(X) models (5), (6) or (7) are called *Kronecker ARX models*, or briefly denoted with KrARX (pronounce *quarks*).

3.1 The identification problem of KrARX models.

The estimation of the matrices $\mathcal{M}_{a,i}$ and $\mathcal{M}_{b,i}$ is unique up to an ambiguity transformation that we describe in the following lemma.

Lemma 3. Let $i \in \{1..p\}$. Denote a low-rank decomposition of $\mathcal{R}(A_i)$ be given by $U_i V_i^T$, where $U_i, V_i \in \mathbb{R}^{N^2 \times r_i}$. Let $\widehat{U}_i, \widehat{V}_i \in \mathbb{R}^{N^2 \times r_i}$.

Then, the pairs of matrices U_i, V_i and $\widehat{U}_i, \widehat{V}_i$ equivalently represent the reshuffled matrix $\mathcal{R}(A_i)$ if and only if they satisfy the following relationships:

$$\widehat{U}_i = U_i T_i, \qquad \widehat{V}_i^T = T_i^{-1} V_i^T$$

where $T_i \in \mathbb{R}^{r_i \times r_i}$ is a non-singular similarity transformation. Moreover, the $N \times N$ matrices in (4) are such that, for $j = 1..r_i$:

$$M(b_i^{(j)}) = ivec\Big(U_i(:,j)\Big)^T, \qquad M(a_i^{(j)}) = ivec\Big(V_i(:,j)\Big)$$

We are now ready to formulate the identification problem. Given the model structure of KrARX models, the problem of identifying these models from measurement sequences $\{S_k\}_{k=1}^{N_t}$ is fourfold:

(1) The temporal order index p.
(2) The spatial order index r_i for each coefficient matrix.
(3) The parametrization of the matrices $M(a_i^{(j)})$ and $M(b_i^{(j)})$. An example of a parametrization of the matrices $M(a_i^{(j)})$ and $M(b_i^{(j)})$ is (block) Banded.
(4) The estimation of the parameter vectors $a_i^{(j)}$, $b_i^{(j)}$ up to an ambiguity transformation. This requires the specification of a cost function. An example of such a cost function using the model (6) is the following least squares cost function,

$$\min_{p, r_i, a_i^{(j)}, b_i^{(j)}} \sum_{k=p+1}^{N_t} \| S_k - \sum_{i=1}^{p} \Big(\sum_{j=1}^{r_i} M(a_i^{(j)}) S_{k-i} M(b_i^{(j)}) \Big) \|_F^2 \tag{9}$$

for data batches with N_t points. By the selection of the parameter p and particular choices of the parametrizations in step 3 above, various special cases of restricting the coefficient matrices A_i in (3) to particular sets (such as $\mathcal{D}^\alpha, \mathcal{D}$ or $\mathcal{K}$) can be considered. Further constraints to the least-squares cost function might be introduced to look for sparsity in the parametrization vectors $a_i^{(j)}$ and $b_i^{(j)}$.

An important challenge of the parameter estimation problem is the *computational efficiency*. The covariance matrix estimation in high dimensional spaces has already been addressed in Tsiligkaridis and Hero (2013) and is not considered further on in this brief paper.

4. ESTIMATING KRARX MODELS WITH A THREE-STAGE APPROACH

In the sequel we assume the Kronecker rank to be time-invariant, i.e $r_i = r$ for all $i \in \{1..p\}$.

4.1 A least squares cost function with rank minimization

According to the model (5) and the definition of the reshuffling operator $\mathcal{R}(.)$ we have,

$$\mathcal{R}\Big(A_i\Big) = \sum_{j=1}^{r} \mathrm{vec}\Big(M(b_i^{(j)})^T\Big)\mathrm{vec}\Big(M(a_i^{(j)})\Big)^T \tag{10}$$

A way to find the spatial order index r is via the Kronecker rank. Let this be denoted by r. Recall that from Lemma 3, the estimation of the matrices $M(a_i^{(j)})$ and $M(b_i^{(j)})$ for different j is not unique. Then, we write

$$\mathcal{R}\Big(A_i\Big) = U_i V_i^T \tag{11}$$

with $U_i, V_i \in \mathbb{R}^{N^2 \times r}$. Not knowing the Kronecker rank, a possible way to retrieve it from data in combination with the coefficient matrices A_i for a given temporal order p is via the following multi-criteria optimization problem:

$$\min_{A_i} \quad \left(\sum_{k=p+1}^{N_t} \| \mathbf{s}_k - \sum_{i=1}^{p} A_i \mathbf{s}_{k-i} \|_F^2, \mathrm{rank}\Big(\mathcal{R}(A_i)\Big) \right) \tag{12}$$

Let the estimated coefficient matrices be denoted by $\widehat{A}_i$, then subsequently an SVD of the matrices $\mathcal{R}\Big(\widehat{A}_i\Big)$ provides

with estimates of the terms U_i and V_i in the Kronecker products in (11). Since the rank operator in the cost function (12) turns this cost function into a non-convex optimization problem, the nuclear norm can be used to convexify this problem. Dealing with nuclear norm regularization on p matrices of size $N^2 \times N^2$ is costly, especially when N is large. However, we see in the following section that some more efficient computations can be performed by parallelizing the optimization problem.

4.2 Local least squares for parallel computations, $r < N$

In this paragraph, we consider $p = 1$ for the sake of clarity. The least squares term in the cost function (12) can be addressed row by row. Let j, n be integers in $\{1, ..., N\}$, and ℓ be such that $\ell = N(n-1)+j$. The ℓ-th line of A_1 is denoted with $\theta_{n,j}$. Using "standard" Matlab notation to select part of a matrix, the matrix $\mathcal{P}(\theta_{i,j})$ and $\mathcal{R}(A_1)$ are related as:

$$\mathcal{P}(\theta_{i,j}) = \mathcal{R}(A_1)(i : N : \text{end}, j : N : \text{end}) \tag{13}$$

To further clarify this notation, we refer to Figure 1 for a display of the different matrices in the above relation.

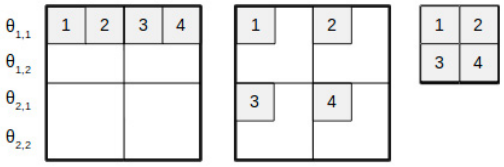

Fig. 1. Schematic representation of the matrices A_1(left), $\mathcal{R}(A_1)$(middle) and $\mathcal{P}(\theta_{1,1})$(right) for a 2-level matrix with 4 blocks.

We state the following lemma:

Lemma 4. If $\text{rank}(\mathcal{R}(A_1)) = r < N$, then for all i, j in the set $\{1..N\}, \text{rank}(\mathcal{P}(\theta_{i,j})) \leq r$.

Proof. Let a low-rank decomposition of $\mathcal{R}(A_1)$ be such that:

$$\mathcal{R}(A_1) = U_i V_i^T$$

$$\mathcal{R}(A_1) = \begin{bmatrix} u_1(1,:) \\ \vdots \\ u_N(1,:) \\ \vdots \\ u_1(N,:) \\ \vdots \\ u_N(N,:) \end{bmatrix} \begin{bmatrix} v_1(1,:)^T & \dots & v_N(1,:)^T & \dots & v_N(N,:)^T \end{bmatrix}$$

where $u_i, v_j \in \mathbb{R}^{N \times r}$. From (13), $\mathcal{P}(\theta_{i,j}) = u_i v_j^T$. The vectors u_i, v_j are not necessarily full column rank, therefore $\text{rank}(\mathcal{P}(\theta_{i,j})) \leq r$. ■

The reverse implication is in general not true. However by assuming that a low-rank decomposition of $\mathcal{P}(\theta_{i,j})$ is given with $u_i v_j^T$ and by estimating all $u_i, v_j \in \mathbb{R}^{N \times r}$, a low-rank matrix $\mathcal{R}(A_1)$ is built using (13).

We describe the algorithm in the following lines. Let the measurement $s_{j,n}(k)$ correspond to the ℓ-th entry of the vector $\mathbf{s}(k)$. Then, a rank-constrained least squares optimization is formulated to estimate the matrix $A_1(\ell, :)$:

$$\min_{A_1(\ell,:)} \sum_{k=2}^{N_t} \|\mathbf{s}_k(\ell) - A_1(\ell,:)\mathbf{s}_{k-i}\|_2^2$$
$$\text{s.t } \text{rank}\Big(\mathcal{P}(A_1(\ell,:))\Big) = r$$

Here we assumed that the row was selected such that the rank constraint can be satisfied. The above problem is convexified as:

$$\min_{A_1(\ell,:)} \sum_{k=2}^{N_t} \|\mathbf{s}_k(\ell) - A_1(\ell,:)\mathbf{s}_{k-1}\|_2^2 + \lambda \|\mathcal{P}(A_1(\ell,:))\|_\star \tag{14}$$

An SVD of the low-rank matrix $\mathcal{P}(A_1(\ell,:))$ yields the following decomposition:

$$\mathcal{P}(A_1(\ell,:)) = [U_{1,\ell} \ \star] \begin{bmatrix} \Sigma_{1,\ell} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{1,\ell}^T \\ \star \end{bmatrix} \tag{15}$$

and then, with $\hat{u}_n = U_{1,\ell}\Sigma_{1,\ell}^{1/2}$ and $\hat{v}_j^T = \Sigma_{1,\ell}^{1/2} V_{1,\ell}^T$

$$\mathcal{P}(A_1(\ell,:)) = \hat{u}_n \hat{v}_j^T \tag{16}$$

where $\hat{u}_n, \hat{v}_j \in \mathbb{R}^{N \times r}$ This decomposition is unique up to a (non-singular) ambiguity transformation $T \in \mathbb{R}^{r \times r}$. Therefore, the SVD in (15) cannot be performed for N independent well-chosen rows, as it would yield N different ambiguity transformations. The remaining column-vectors of both U_i, V_i ($2(N-1)$ matrices of size $N \times r$) are rather estimated via least squares based on the estimates $\hat{u}_n, \hat{v}_j$. If (14) is solved e.g for $\ell = 1$, then for $n = 1$ and j in the set $\{2, ..., N\}$, we solve the constrained least-squares optimization:

$$\min_{\theta_{1,j}, v_j} \sum_{k=2}^{N_t} \|\mathbf{s}_k(\ell) - \theta_{1,j}\mathbf{s}_{k-1}\|_2^2$$
$$\text{s.t } \mathcal{P}(\theta_{1,j}) = \hat{u}_1 v_j^T$$

that is equivalently parametrized with only N unknowns:

$$\min_{v_j} \quad \sum_{k=2}^{N_t} \|\mathbf{s}_k(\ell) - \mathcal{M}(\hat{u}_1, v_j)\mathbf{s}_{k-1}\|_2^2 \tag{17}$$

where $\mathcal{M}(.,.)$ is the operator defined with:

$$\mathcal{M}(\hat{u}_1, v_j) = \begin{bmatrix} \hat{u}_1(1,:)v_j^T & \dots & \hat{u}_1(N,:)v_j^T \end{bmatrix}$$

Similarly, for $j = 1$ and n in the set $\{2, ..., N\}$:

$$\min_{u_n} \quad \sum_{k=2}^{N_t} \|\mathbf{s}_k(\ell) - \mathcal{M}(u_n, \hat{v}_1)\mathbf{s}_{k-1}\|_2^2 \tag{18}$$

These $2(N-1)$ least squares can be performed in parallel, each of which corresponds to one sensor location as can be visualized in Figure 2. Choosing the adequate sensor location, i.e ($j = 1, n = 1$) in (14), is not unique. We summarize the algorithm with the following three steps:

(1) Solve the nuclear-norm regularized optimization in (14), e.g $\ell = 1$.
(2) Compute the SVD of $\mathcal{P}(A_1(\ell,:))$ and estimate $\hat{u}_1, \hat{v}_1$.
(3) Solve, in parallel, the least squares:

$$n = 1, \forall j \in \{2, ..., N\}, \text{ solve (17)}$$
$$j = 1, \forall n \in \{2, ..., N\}, \text{ solve (18)}$$

4.3 Generalization to higher Kronecker ranks

The three-step algorithm has been discussed for the case $r < N$. In most cases, the Kronecker rank is not known *a priori* and it has to be detected with cross-validation. Therefore we analyze how to deal with higher Kronecker ranks, more particularly $r < 2N$. In the previous paragraph, the Kronecker rank was limited by the size of the submatrices $\mathcal{P}(\theta_{i,j}) \in \mathbb{R}^{N\times N}$. A submatrix of size $2N\times 2N$ shall be selected such that the SVD in (15) is then carried out on a rank-deficient matrix L. For example, when considering the output data at the locations $\{(n,j)\}_{n,j=1,2}$, the matrix L is defined as:

$$L = \begin{bmatrix} \mathcal{P}(\theta_{1,1}) & \mathcal{P}(\theta_{1,2}) \\ \mathcal{P}(\theta_{2,1}) & \mathcal{P}(\theta_{2,2}) \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} v_1^T & v_2^T \end{bmatrix}$$

Let us denote the set $\{\ell \in \mathbb{N} | \ell = N(n-1)+j, (n,j) \in \{1,2\}\}$ with $\mathcal{L}$. The regularized least squares in (14) is then extended into:

$$\min_{A_1(\ell,:)} \quad \sum_{k=2}^{N_t} \sum_{\ell\in\mathcal{L}} \|\mathbf{s}_k(\ell) - A_1(\ell,:)\mathbf{s}_{k-1}\|_2^2 + \lambda \|L\|_\star \qquad (19)$$

Similarly, the SVD is performed on L and the least squares in (17) and (18) are formulated using the estimates $\{\hat{u}_i, \hat{v}_i\}_{i=1,2}$. Figure 2 illustrates.

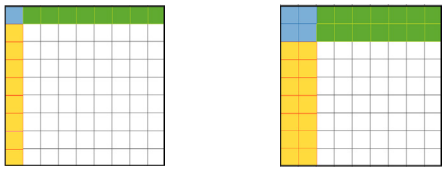

Fig. 2. Square array: the colored entries locate in the matrix S_k which data to consider in order to estimate the coefficient-matrices with minimum computational complexity. Blue entry: minimization of type (14)-(16). Yellow entry: minimization of type (17). Green entry: minimization of type (18). Left: identification for matrices of Kronecker rank smaller than N. Right: Kronecker rank between N and $2N$.

The computational complexity is reduced exploiting the Kronecker structure and is attractive for being non-iterative and parallelizable to a large extent.

4.4 Computational complexity

Consider the estimation problem (14) with $\lambda = 0$, $N_t = N$ and $r < N$. Solving the full unstructured least squares requires inverting a $N^2 \times N^2$ matrix and a matrix-matrix multiplication, which scales up to $\mathcal{O}(2N^6)$. Solving (14) for only the least-squares term requires inverting a matrix of size $N^2 \times N^2$. Moreover, the SVD (16) is performed on a matrix of size $N \times N$. Last, each of the $2(N-1)$ least-squares with N unknowns in (17) and (18) scales up to $\mathcal{O}(N^3)$. They can be carried out in parallel with η different workers, hence the complexity $\mathcal{O}(\frac{2(N-1)N^3}{\eta})$ for this third step.

Another optimization method using Alternating Least Squares and featuring only $2rN^2$ unknowns is proposed in Sinquin and Verhaegen (2016).

5. ILLUSTRATIVE EXAMPLE

In this numerical illustration, we consider an 2D array of subsystems that interact with their neighbors through some unknown coupling. The spatial and temporal behavior of each subsystem is stable and all subsystems are assumed identical. We introduce a state-space representation of the network with the 2-level Sequentially Semi-Separable structure. Such a 2D model is built from a 1D string of 1D strings. A SSS matrix $\bar{M}$ is built from a set of matrices-generators as follows:

$$\bar{M} = \begin{bmatrix} D & UV & \dots & UW^{N-2}V \\ PQ & D & & \vdots \\ \vdots & & \ddots & \vdots \\ \vdots & \vdots & \ddots & UV \\ PR^{N-2}Q & \dots & \dots & PQ \quad D \end{bmatrix}$$

which is compactly denoted with:

$$\bar{M} = \mathcal{SSS}(P,R,Q,D,U,W,V)$$

Both R and Q have spectral radius strictly inferior to 1. The matrix $\bar{M}$ represents the static input-output map from a string of N mixed causal anti-causal interconnected systems. A 2-level matrix $\mathbf{M}$ is built from N sets of 1D SSS matrices as generators:

$$\mathbf{M} = \mathcal{SSS}(\bar{P},\bar{R},\bar{Q},\bar{D},\bar{U},\bar{W},\bar{V})$$

More details on such recursive structures can be found in Rice (2010). Although each of the 1D matrix generator (such as $\bar{P} \in \mathbb{R}^{N\times N}$) are block-Toeplitz, it is no longer the case for the 2-level matrix as the product of two block-Toeplitz matrices is in general not block-Toeplitz.

For $\mathbf{A}, \mathbf{B}, \mathbf{C}$ defined as 2-level SSS matrices, we write the global state-space:

$$\begin{cases} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{e}_k \end{cases} \qquad (20)$$

where:

$$\mathbf{A} \in \mathbb{R}^{N^2n^2\times N^2n^2}, \mathbf{B} \in \mathbb{R}^{N^2n^2\times N^2m}, \mathbf{C} \in \mathbb{R}^{N^2p\times N^2n^2}$$

We choose $N = 10, n = 2, m = 1, p = 1$. In this section, we estimate a high-order Finite Impulse Response (FIR) model from input-output data generated according to the state-space (20). The noise sequence $\mathbf{e}_k$ is a white Gaussian noise with zero mean and a variance such that the Signal to Noise Ratio on the output is 20dB. The temporal order of the FIR model is chosen equal to 5. The identification and validation set are independent from one another and each contain 5×10^3 temporal measurements.

The coefficient-matrices $\{A_i\}_{i=1..p}$ (corresponding to the Markov parameters $\{\mathbf{C}\mathbf{A}^{i-1}\mathbf{B}\}_{i=1..p}$) are estimated using the following methods:

- a standard unstructured least-squares

$$\min_{A_i} \sum_{k=p+1}^{N_t} \|\mathbf{y}_k - \sum_{i=1}^{p} A_i \mathbf{y}_{k-i}\|_2^2$$

- the KrARX identification with (14), (17) and (18). The Kronecker rank takes values between 1 and 10.
- a sparse least-squares solving the following minimization, Kim et al. (2007):

$$\min_{A_i} \sum_{k=p+1}^{N_t} \|\mathbf{y}_k - \sum_{i=1}^{p} A_i \mathbf{y}_{k-i}\|_2^2 + \mu \sum_{i=1}^{p} \|\mathrm{vec}\Big(A_i\Big)\|_1$$

where μ belongs to the range $logspace(1, 4, 5)$.

The relative Root-Mean-Square-Error (RMSE) between the output signals $\mathbf{y}$ and the prediction $\widehat{\mathbf{y}} := \sum_{i=1}^{p} \widehat{A}_i \mathbf{y}_{k-i}$ is defined with:

$$\text{Relative RMSE}(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{\sum_{k=1}^{N_t} \|\mathbf{y}_k - \widehat{\mathbf{y}}_k\|_2^2}{\sum_{k=1}^{N_t} \|\mathbf{y}_k\|_2^2}$$

where N_t is the number of temporal samples in the validation set.

We define the *storage complexity* as the number of non-zero entries needed to construct the p coefficient-matrices. For example the storage of a KrARX model requires at most $2prN^2$ entries, i.e only the non-zero elements of the factor matrices, while it reaches a total of pN^4 for the full least squares estimation. It is illustrated in Figure 3 that displays the relative RMSE with respect to the 0-norm of the entries needed to construct the full coefficient matrix.

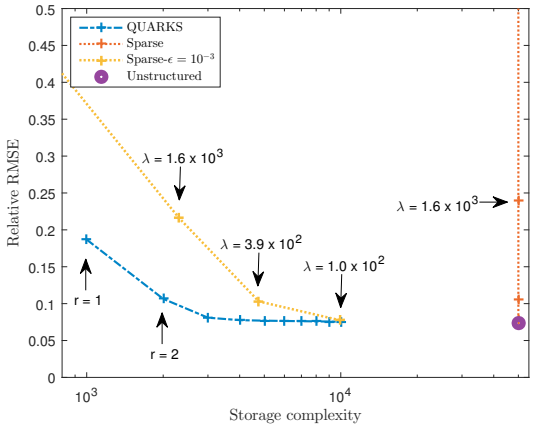

Fig. 3. Relative RMSE for the output error on validation data versus the storage complexity.

The *exact* number of non-zero entry (computed with the Matlab function *nnz*) stays constant with increasing weight on the sparse prior. Truncating all values from the matrices estimated with ℓ_1 prior below 10^{-6} and then computing the Relative RMSE on validation data yields a better data-sparse representation that is however outperformed by the Kronecker representation in terms of Relative RMSE.

Consequently, the ability for the KrARX to model MSSS structures with few Kronecker terms is related to first, both the stability of the global system (small temporal order p), and second, the approximate block-Toeplitz with Toeplitz-block structure from the coefficient-matrices.

6. CONCLUSION

In this paper the class of Kronecker networks is defined and the modeling of temporal dynamics with VAR models is investigated. Each coefficient-matrix of the VAR model is approximated with a sum of few Kronecker matrices which offers high data compression for large networks. Estimating in least-squares sense the data matrices gives rise to a bilinear problem which is addressed with a three-stage method. Numerical examples on a network modeled with Multi-level Sequentially Semi-Separable matrices demonstrates comparable output-error performances with the unstructured least-squares with a low-Kronecker rank representation which enables a more efficient storage of the entries for large coefficient-matrices than with a sparse prior. Such a modeling opens up the way for more useful identification algorithms, e.g subspace identification.

For an alternative solution of the bilinear estimation problem via Alternating Least Squares and an application to atmospheric turbulence modeling, we refer to Sinquin and Verhaegen (2016). The stability of the KrARX model identified has not been presented in this paper and is subject of current investigations.

REFERENCES

Hansen C., Nagy J. G. and O'Leary D. P. Deblurring images : matrices, spectra, and filtering. *Fundamentals of Algorithms 3, SIAM, Philadelphia*, 2006.

Kim S.-J., Koh K., Lustig M., Boyd S., Gorinevsky D. An Interior-Point Method for Large-Scale l1-Regularized Least Squares. *IEEE Journal on Selected Topics in Signal Processing, 1(4):606-617*, December 2007.

Loan, C.F. van,. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, volume 123, pages 85–100, 2000.

Loan, C.F. van, and Vokt J.P. Approximating matrices with multiple symmetries. *Siam J. Matrix Anal. Appl.*, volume 36, pages 974–993, 2015.

Loan, C.F. van, and Pitsianis, N.P. Approximation with Kronecker products. *Linear Algebra for Large Scale and Real Time Applications*, Kluwer Publications, Dordrecht, volume 25, pages 293–314, 1992.

Massioni, P. Distributed control for alpha-heterogenous dynamically coupled systems. *Systems & Control Letters*, volume 72, pages 30–35, 2014.

Rice, J.K. Efficient algorithms for distributed control: a structured matrix approach. *PhD thesis, Technische Universiteit Delft*, 2010.

Roggemann M. C. and Welsh B. Imaging Through Turbulence *CRC Press, Boca Raton, FL*, 1996.

Sinquin, B., and Verhaegen, M. Kronecker-based modeling of networks with unknown communication links. *arXiv:1609.07518 [cs.SY]*, 2016.

Tsiligkaridis, T. and Hero, A.O. Covariance Estimation in High Dimensions Via Kronecker Product Expansions. *IEEE Transactions on Signal Processing*, volume 61, 2013.

Udell, M., Horn, C., Zadeh, R. and Boyd, S. Generalized Low Rank Models. *Foundations and Trends in Machine Learning*, volume 9, pages 1–118, 2016.

Wang, J., Zhang, Q., Ljung, L. Revisiting Hammerstein System Identification through the Two-Stage Algorithm for Bilinear Parameter Estimation. *Automatica*, volume 45, no. 11, pages 2627–2633, 2001.

Zorzi, M., and Chiuso, A. Sparse plus Low rank Network Identification: A Non-parameteric Approach. *Automatica - accepted*, 2016.