Recursive Kronecker-based Vector AutoRegressive identification for large-scale adaptive optics

Monchen, Guido; Sinquin, Baptiste; Verhaegen, Michel

DOI
10.1109/TCST.2018.2834521

Publication date
2018

Document Version
Final published version

Published in
IEEE Transactions on Control Systems Technology

Recursive Kronecker-Based Vector Autoregressive Identification for Large-Scale Adaptive Optics

Guido Monchen, Baptiste Sinquin, and Michel Verhaegen

***Abstract*— This brief presents an algorithm for the recursive identification of Vector AutoRegressive (VAR) models of large dimensions. We consider a VAR model where the coefficient matrices can be written as a sum of Kronecker products. The algorithm proposed consists of recursively updating the Kronecker factor matrices at each new time step using alternating least squares. When the number of terms in the Kronecker sum is small, a significant reduction in computational complexity is achieved with respect to the recursive least squares algorithm on an unstructured VAR model. Numerical validation of nonstationary atmospheric turbulence data, both synthetic and experimental, is shown for an adaptive optics application. Significant improvements in accuracy over batch identification methods that assume stationarity are observed while both the computational complexity and the required storage are reduced.**

***Index Terms*— Kronecker product, large-scale systems, recursive least-squares (RLSs), system identification, vector autoregressive (VAR).**

I. Introduction

IDENTIFYING large-scale time-varying linear systems is a major interest in a number of applications such as Adaptive Optics (AO) for astronomy applications [1]. Such systems reshape a wavefront in real time so as to compensate for the disturbances caused by the light having to travel through the turbulent atmosphere [2]. Recent technological advancements in increasing the size of the prime mirror in ground-based telescopes result in systems with thousands of sensors and actuators [3] that require scalable identification and control algorithms. Although state-space modeling and $\mathcal{H}_2$ optimal control for AO have shown significant improvements in the control performances over nonpredictive methods [4], and scaling up the number of sensor channels has so far been a challenge for both identification and control. Guyon and Males [5] analyze the use of Vector AutoRegressive (VAR) models to accurately predict the wavefront based on pseudo open-loop wavefront data. However, the method does not scale well as it does not exploit the structure on the coefficient matrices to allow its effective use for a large-scale AO. The dimensionality prevents obtaining both accurate and data-sparse models. In this brief, the analyzed input–output representation is the VAR model [6]. When sensor measurements are available on a square array of size $N \times N$ for N_t time samples, the coefficient matrices belong to $\mathbb{R}^{N^2 \times N^2}$ and the computational complexity for the estimation in least-squares sense reaches $\mathcal{O}(N^6 N_t)$. Structural assumptions on the matrices were studied in order to obtain the best tradeoff between the compactness of the model and its accuracy. First, the spatial coupling in the turbulence is ignored when modeling the coefficient matrices with a *diagonal* structure as discussed in [7]. In [8], the identification of *sparse* models has been investigated but requires the knowledge of the topology of the network, which may not be available beforehand. Such sparsity patterns may be too restrictive for modeling the atmospheric turbulence without increasing the variance of the prediction error significantly with respect to the unstructured case. There is indeed no guarantee that the spatial dynamics of the turbulence considered in AO applications exhibit such a sparse structure.

When modeling 2-D spatial systems with temporal dynamics, a new class of Kronecker-VAR models named QUARKS is introduced in [9], which parameterizes each coefficient matrix as a sum of Kronecker products [10]. The QUARKS model features matrices of size $N \times N$ rather than $N^2 \times N^2$ in the unstructured VAR model. Hence, the modeling paradigm enables high data compression when there are only a few terms r in the Kronecker sum, while achieving similar prediction error performances with respect to the unstructured estimation. Estimating QUARKS scales with $\mathcal{O}(r N^3 N_t)$. Sinquin and Verhaegen [9] consider the use of QUARKS models on stationary turbulence; however, nonstationary turbulence has not yet been analyzed in this modeling class.

The use of recursive algorithms for handling the identification of large-scale systems has two interests: 1) to cope with temporal variations in the data and 2) to reduce the computational complexity once an initial guess has been obtained using a batch method. Updating the coefficient matrices recursively in an unstructured VAR model is nonetheless still not computationally feasible for large-scale systems as it scales with $\mathcal{O}(N^4)$. Both diagonal and block banded with banded blocks structures enable a decentralized and scalable recursive identification using the standard recursive least-squares (RLS) algorithm [11] locally. In this brief, we consider the modeling of large-scale 2-D spatial–temporal systems that have varying dynamics and propose a recursive identification of QUARKS models. More specifically, we investigate how to

Manuscript received January 22, 2018; revised April 23, 2018; accepted May 2, 2018. Manuscript received in final form May 6, 2018. 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. Recommended by Associate Editor X. Chen. *(Corresponding author: Baptiste Sinquin.)*

The authors are with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: guido.monchen@gmail.com; baptiste.sinquin@gmail.com; m.verhaegen@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2018.2834521

recursively update the coefficient matrices of the QUARKS model in order to deal with 2-D spatial systems that are affected by nonstationary disturbances. While the unstructured VAR model has coefficient matrices that can be estimated in a convex large-scale optimization problem, the matrix-VAR model with Kronecker modeling is bilinear and nonconvex. Sinquin and Verhaegen [9] propose an alternating least squares (ALS) algorithm that estimates the coefficient matrices iteratively and converges to the global minimum under certain conditions. To reduce the variance of the prediction error for nonstationary disturbances, the RLS algorithm is analyzed to keep the coefficient matrices of the QUARKS model up-to-date [12]. Apart from improving over the standard identification of QUARKS for nonstationary disturbances, the recursive algorithm also significantly reduces the storage required for estimating the QUARKS models for which no input data is discarded, contrary to the case where a sparse pattern is assumed on the coefficient matrices.

This brief is organized as follows. In Section II, we introduce the QUARKS models and review how to efficiently identify them using ALS. In Section III-A, we introduce the method to update these models recursively. The computational complexity between the unstructured RLS and the proposed algorithm is compared in Section IV, and a numerical validation in the field of AO with synthetic and laboratory testbed data is given in Section V.

Notations: A vector variable is written in lower case letters (x) and matrix variables are written in upper case letters (X). The Kronecker product between two matrices A and B is denoted as $A \otimes B$. The vec(X) operator performs the vectorization of the matrix X, thus stacking all columns below each other. The two-norm of a vector x is denoted as $\|x\|_2$, and the Frobenius norm of a matrix X is written as $\|X\|_F$. The big-O notation is used for describing computational complexities and indicates the asymptotic growth rate of the computational cost for a given mathematical operation. We moreover recall that $y = (B \otimes C)x \equiv Y = \mathrm{CXB}^T$, where $x = \mathrm{vec}(X)$ and $y = \mathrm{vec}(Y)$.

II. Modeling 2-D Spatial–Temporal Systems With QUARKS

Let the measurements at time instant k be sampled from a regular grid such that $Y(k) \in \mathbb{R}^{m \times n}$. We denote the lifted vector with $y(k) = \mathrm{vec}(Y(k))$ and consider the following VAR model with temporal order p:

$$y(k) = \sum_{i=1}^{p} A_i y(k-i) + v(k) \tag{1}$$

where $v(k) \in \mathbb{R}^{mn}$ is a zero-mean white noise process with the covariance matrix $R_v = I$, N_t is the number of time samples considered, and $p < k \leq N_t$. The identification problem for this model is to estimate the coefficient matrices $A_i \in \mathbb{R}^{mn \times mn}$ using all available data up to time N_t. We assume, following the lines in [9], that we can parametrize the matrices A_i as a sum of a few Kronecker products

$$A_i = \sum_{j=1}^{r_i} \left(B_i^{(j)}\right)^T \otimes C_i^{(j)} \tag{2}$$

where r_i is the so-called Kronecker rank of the matrix A_i. The Kronecker rank defines the number of terms used in the sum to construct A_i from $B_i^{(j)} \in \mathbb{R}^{n \times n}$ and $C_i^{(j)} \in \mathbb{R}^{m \times m}$. To achieve an efficient representation for the matrices A_i, we assume $r_i \ll mn$. The low-Kronecker rank decomposition of the coefficient matrices as described in [9] is especially interesting when there are only a few terms in the Kronecker sum of factor matrices and when the coefficient matrices are dense. In other words, the more sparse is, the less advantageous the Kronecker structure is with respect to the sparse structure.

The VAR model is rewritten as

$$Y(k) = \sum_{i=1}^{p} \left(\sum_{j=1}^{r_i} C_i^{(j)} Y(k-i) B_i^{(j)} \right) + V(k) \tag{3}$$

where $V(k) = \mathrm{vec}(v(k))$. The model (3) is called as *Kronecker VAR model* and is abbreviated as QUARKS [9]. The problem of identifying the coefficient matrices now changes to identifying the matrices $B_i^{(j)}$ and $C_i^{(j)}$. Let us write the identification as the following minimization problem:

$$\min_{B_i^{(j)}, C_i^{(j)}} \sum_{k=p+1}^{N_t} \left\| Y(k) - \sum_{i=1}^{p} \left(\sum_{j=1}^{r_i} C_i^{(j)} Y(k-i) B_i^{(j)} \right) \right\|_F^2. \tag{4}$$

This minimization problem is bilinear in its unknowns and as such, we cannot write a closed-form solution for it. One way to solve this is to use an algorithm based on the nonlinear Gauss–Seidel method. Each iteration of this ALS method alternatively estimates the matrices $B_i^{(j)}$ and $C_i^{(j)}$ until the convergence is at a minimum. In [9], the convergence to the global minimum of the cost function was proven using the fixed point theory when the columns of one factor matrix were normalized. However, normalization requires the knowledge of the true factor matrix, which is not available in practice. Since this algorithm is aimed at the practical application to AO, the normalization step is skipped. For N_t sufficiently large, an empirical convergence to the global minimum for a general temporal order and the Kronecker rank was observed without normalizing, starting from random initial guesses and with persistently exciting data. The two latter conditions ensure that each step in the ALS has a unique solution. For example, if the initial guesses are zero, then the regression matrix is not invertible and the iterations cannot start. It is shown in Section V that removing the normalization slightly increases the convergence speed and overall accuracy of the model.

Denote a prior nonzero initial guess of the matrix C as $\widehat{C} = C(0)$ and N_a as the total number of alternating iterations performed. When choosing for simplicity $p = r_i = 1$, the ALS algorithm for estimating the coefficient matrices B, and C consists of the steps highlighted in Algorithm 1.

III. Online Updating of QUARKS

A. Recursive Least Squares for Updating Unstructured VAR Models

The ALS algorithm provides a computationally efficient way for the offline estimation of QUARKS parameters on a prior identification data set. Because of the nonstationary

Algorithm 1 QUARKS($\{y(k)\}_{k=1:N_t}, n, m, N_a, e_{min}$)

Input : $\{y(k)\}_{k=1:N_t}, n, m, N_a, e_{min}$
Output: $\widehat{B}, \widehat{C}$

$\ell = 0$, $e = +\text{Inf}$
$\widehat{C} = \text{randn}(m, m)$
while $\ell < N_a$ *and* $e > e_{min}$ **do**
  Solve, for all $j = 1 \dots n$,
  $\min_{b_j} \sum_{k=1}^{N_t} ||Y(k)(:, j) - \widehat{C}Y(k-1)b_j||_2^2$
  Denote the solution with $\widehat{B}$
  Solve, for all $j = 1 \dots m$,:
  $\min_{c_j} \sum_{k=1}^{N_t} ||Y(k)(j, :) - c_j^T Y(k-1)\widehat{B}||_2^2$
  Denote the solution with $\widehat{C}$
  $\epsilon(k) = \sum_{k=1}^{N_t} ||Y(k) - \widehat{C}Y(k-1)\widehat{B}||_F^2$
  $e = |\epsilon(k) - \epsilon(k-1)|$
  $\ell = \ell + 1$
end

nature of the atmospheric disturbances in AO applications, we investigate whether the algorithm can be implemented in a recursive manner. In this way, the coefficient matrices can be continuously updated during closed-loop operation of the system. For unstructured VAR models like in (1), we focus on the RLS algorithm [12], [13].

We first discuss the RLS algorithm in the unstructured VAR case (1) for the case $p = 1$. Let us consider at time instant k to have an estimate of the parameter matrix A, denoted with $\widehat{A}(k)$. The variables $\hat{a}_j(k)$ for all $j = 1, \dots, mn$ denote the rows of the matrix $\widehat{A}$. The jth element of the vector $y(k)$ is denoted with $y_j(k)$. Whenever a new measurement $y(k+1)$ becomes available, these estimates are updated. Such an update is the fusion of the prior information and the information about A derived from the new measurements. This fusion can be interpreted as optimizing the following cost function for all the rows a_j of A:

$$\forall\ j = 1, \dots, mn, \quad \min_{a_j}\ \lambda[a_j - \hat{a}_j(k)]P_j(k)^{-1}[a_j - \hat{a}_j(k)]^T + \left(y_j(k+1) - y(k)^T a_j^T\right) R_v^{-1} \left(y_j(k+1) - y(k)^T a_j^T\right) \tag{5}$$

where λ is a forgetting factor in the interval $[0, 1]$, and $P_j(k)$ represents the covariance matrix defined as

$$P_j(k) = \mathbb{E}[(a_j - \hat{a}_j(k))^T (a_j - \hat{a}_j(k))].$$

Equation (5) is equivalently written as

$$\forall\ j = 1, \dots, mn \quad \min_{a_j}\ \mu^T \mu \text{ s.t: } \begin{bmatrix} y_j(k+1) \\ \hat{a}_j^T(k) \end{bmatrix} = \begin{bmatrix} y(k)^T \\ I \end{bmatrix} a_j^T + \begin{bmatrix} R_v^{1/2} & 0 \\ 0 & \lambda^{-1/2} P_j(k)^{1/2} \end{bmatrix} \mu(k) \tag{6}$$

with $\mathbb{E}[\mu(k)\mu(k)^T] = I$ and $\mathbb{E}[\mu(k)] = 0$. The solution to this least squares problem is given by the following recursive equations:

$$\begin{aligned} \hat{a}_j(k+1)^T &= \hat{a}_j(k)^T + g_j(k+1)[y_j(k+1) - y(k)^T \hat{a}_j(k)^T] \\ g_j(k+1) &= \lambda^{-1} P_j(k) y(k) [R_v + \lambda^{-1} y(k)^T P_j(k) y(k)]^{-1} \\ P_j(k+1) &= \lambda^{-1} [P_j(k) - g_j(k+1) y(k)^T P_j(k)]. \end{aligned} \tag{7}$$

We summarize the algorithm updating the estimates of $a_j(k)$ and $P(k)$ in a computationally efficient manner in Algorithm 2. When $P_j(0)$ is chosen identical for all j, then $g_j(k)$ and $P_j(k)$ are independent of j and can be written as $g(k)$ and $P(k)$.

Algorithm 2 RLS($y(k+1)$, $y(k)$, $\widehat{A}(k-1)$, $P(k-1)$, λ, N)

Input : $y(k+1), y(k), A(k-1), P(k-1), \lambda$
Output: $\widehat{A}(k), P(k)$

1 $g(k) = \lambda^{-1} P(k-1) y(k) [I + \lambda^{-1} y(k)^T P(k-1) y(k)]^{-1}$
2 $P(k) = \lambda^{-1} [P(k-1) - g(k) y(k)^T P(k-1)]$
3 **for** $j = 1 \dots N$ **do**
4 $\hat{a}_j(k)^T = \hat{a}_j(k-1)^T + g(k)[y_j(k+1)^T - y(k)^T \hat{a}_j(k-1)^T]$
5 **end**

The solution to (5) can thus be written as

$$[\widehat{A}(k), P(k)] = \textbf{RLS}(y(k+1), y(k), \widehat{A}(k-1), P(k-1), \lambda, mn).$$

Transposing the scalar value $y_j(k+1)$ on line 4 has no effect here; however, in Section III-B, this value will be a vector that makes the transposition necessary. The initial estimate for A can be determined by doing an initial offline identification step or can be set to a random matrix. The initial value for the matrix P is usually set to δI, where I is the identity matrix of appropriate size and δ is a design parameter. Choosing δ depends on how much confidence is placed in the initial guess of A, e.g., a low value for δ means that a high amount of confidence in the initial guess.

B. Recursive Least Squares for QUARKS Models

The algorithm in Section III-A gives an update at each new time step k for the coefficient matrix A of the VAR model in (1), thus providing the possibility to update this coefficient matrix each time when new data are available. We now address the question whether this scheme can be adapted to recursively update a QUARKS model. In particular, there are two matrices that need to be updated, thus creating a bilinear minimization problem with no closed-form solution. A similar problem was tackled for estimating bilinear systems in the case where B and C are the vectors in [14].

For the QUARKS model in (3) with $p = r_i = 1$

$$Y(k) = CY(k-1)B + V(k). \tag{8}$$

Let us denote the estimates for B and C at time k as $\widehat{B}(k)$ and $\widehat{C}(k)$. In this brief, we propose to use the initial estimate of the matrix C, denoted as $\widehat{C}(k)$, and update the estimate of B, denoted as $\widehat{B}(k+1)$. This is then followed by updating C by fixing the previously obtained estimate for B, resulting

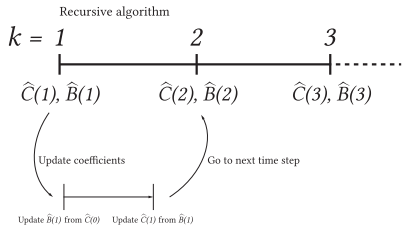

Fig. 1. Time line of the QUARKS-RLS algorithm. Every time step k, we calculate estimates $\widehat{C}(k)$ and $\widehat{B}(k)$ using two alternating steps.

in one ALS update. A schematic is presented in Fig. 1. For each time step k, the factor matrices B and C are updated once.

Similar to Section III-A, we partition $\widehat{B}(k)$, $\widehat{C}(k)$, and $Y(k)$ as follows:

$$\widehat{C}(k) = \begin{bmatrix} \hat{c}_1(k) \\ \vdots \\ \hat{c}_m(k) \end{bmatrix} \quad \widehat{B}(k) = [\hat{b}_1(k) \quad \cdots \quad \hat{b}_n(k)]$$

$$Y(k) = [y_1(k) \quad \cdots \quad y_n(k)] = \begin{bmatrix} \bar{y}_1(k) \\ \vdots \\ \bar{y}_m(k) \end{bmatrix}.$$

The variables $y_i(k)$, $i = 1, \ldots, n$ and $\bar{y}_i(k)$, $i = 1, \ldots, m$ are now the vectors instead of scalars as in Section III-A. We moreover introduce the new variable $U_c(k)$ as

$$U_c(k) = \widehat{C}(k)Y(k)$$

and consider the following problem for updating the estimate of the columns $b_j(k)$ of the matrix $B(k)$ using $\widehat{C}(k)$ inspired by the solution of the previous section

$$\forall\ j = 1, \ldots, n \quad \min_{b_j} \lambda^{-1}[b_j - \hat{b}_j(k)]^T P_b^{-1}(k)[b_j - \hat{b}_j(k)] + \|y_j(k+1) - U_c(k)b_j\|_2^2 \qquad (9)$$

where we consider the case $R_{v_j} = I$. The solution to (9) is provided in Algorithm 2 and can be written as

$$[\widehat{B}(k), P_b(k)] = \mathbf{RLS}(Y(k+1), U_c(k), \widehat{B}(k-1), P_b(k-1), \lambda, n).$$

The second step of the ALS consists of updating $C(k)$ based on the estimate $\widehat{B}(k)$

$$\forall j = 1, \ldots, m \quad \min_{c_j} \lambda^{-1}[c_j - \hat{c}_j(k)]P_c^{-1}(k)[c_j - \hat{c}_j(k)]^T + \left\|\bar{y}_j(k+1)^T - U_b(k)c_j^T\right\|_2^2 \qquad (10)$$

where

$$U_b(k) = \widehat{B}(k)^T Y(k)^T.$$

The solution to (10) is obtained by running Algorithm 2 with the following parameters:

$$[\widehat{C}(k)^T, P_c(k)] = \mathbf{RLS}(Y(k+1)^T, U_b(k), \widehat{C}(k-1)^T, P_c(k-1), \lambda, m).$$

Applying the ALS algorithm for updating the matrices $\widehat{B}(k)$ and $\widehat{C}(k)$ in the minimization problems in (9) and (10) results in the RLS algorithm for QUARKS models as defined in Algorithm 3.

Algorithm 3 QUARKS-RLS Algorithm

$P_b(0) = \delta I$
$P_c(0) = \delta I$
for $1 \leq k < N_t$ **do**
  $U_c(k) = \widehat{C}(k)Y(k)$
  $[\widehat{B}(k), P_b(k)] =$ **RLS**$(Y(k+1), U_c(k), \widehat{B}(k-1), P_b(k-1), \lambda, n)$
  $U_b(k) = \widehat{B}(k)^T Y(k)^T$
  $[\widehat{C}(k)^T, P_c(k)] =$ **RLS**$(Y(k+1)^T, U_b(k), \widehat{C}(k-1)^T, P_c(k-1), \lambda, m)$
end

By performing these steps at each new time step k, we obtain an RLS algorithm for low Kronecker-rank-structured models. The initial guess for the coefficient matrices $\widehat{B}(0)$ and $\widehat{C}(0)$ is obtained using the QUARKS as described in Section I and [9] for more thorough explanations.

IV. Computational Complexity

We compare the computational complexity of the recursive algorithm for the QUARKS model with the unstructured model and distinguish between the offline and online computational parts. The offline part has no real-time constraints, although recursive algorithms do not store the whole data batch but only the last measurement and the matrices P_b and P_c. This makes it attractive even for offline use on large-scale stationary data. Of higher interest is the online computational complexity, as it is constrained by the frequency of operation of the system.

When considering the RLS equations in Algorithm 2 without the Kronecker structure, the most computationally complex operation is a matrix–vector multiplication $[P(k)y(k)]$. The complexity is $\mathcal{O}(m^2n^2)$. For the RLS equations using the Kronecker structured matrices in Algorithm 3, the most complex operation is a matrix–matrix multiplication (MMM) $[P_b(k)Y(k)]$. Its computational complexity is $\mathcal{O}(m^2n)$ if $m > n$ or $\mathcal{O}(mn^2)$ if $m < n$.

V. Numerical Validation

The algorithm is validated using an application to AO using synthetic and validation data. The light from a star needs to pass through the Earth's atmosphere before it can be seen by a ground-based telescope. The wavefront that enters the atmosphere is flat and smooth while it is no longer the case when it reaches the aperture of the telescope. In order to restore the wavefront to its original shape, a combination of a wavefront sensor (WFS) and a deformable mirror is used to compensate for this disturbance. The WFS consists of an array of small lenslets which focuses the light for each lenslet in a point on an imaging sensor. Because of the local curvature of the wavefront, these points deviate from their original position (i.e., when the wavefront is flat). The amplitude of the deviation in the horizontal and vertical coordinates yields

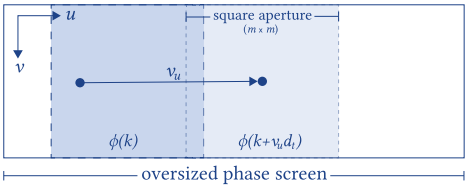

Fig. 2. Generating an oversized phase screen over which a smaller square aperture will move. By varying the speed v_u at which this aperture will move, the simulated wind speed is changed.

two measurements of the local derivative of the wavefront. In this particular example, the focus is on an open-loop model.

A. Synthetic Data

The turbulence is generated according to the multiscale phase screen synthesis approach detailed in [15]. More specifically, only the low-resolution process is used here based on the fast Fourier transform moving average (FFT-MA) generator [16]. In short, the phase screen x with dimensions $m \times m$ can be represented as an MA model

$$x(u,v) = \sum_{k_u,k_v} \theta(k_u,k_v)\epsilon(u-k_u, v-k_v) \tag{11}$$

with ϵ being a zero-mean white noise process with variance 1, and θ being the MA coefficients. To determine the MA coefficients θ, the spatial covariance matrix $C_\phi(u,v)$ of the atmospheric turbulence based on the von Karman theory [1] is considered, such that

$$C_\phi(u,v) = \sum_{k_u,k_v} \theta(k_u,k_v)\theta(u+k_u, v+k_v). \tag{12}$$

The coefficients θ can now be calculated from the spatial covariance matrix C_ϕ using the FFT-MA generator. Since $C_\phi(u,v)$ tends to zero for large u, v, the index terms k_u, k_v can be seen as finite and are assumed to be $k_u = -\delta, \ldots, \delta$ and $k_v = -\delta, \ldots, \delta$.

The wind speed is simulated by generating an oversized turbulence phase screen and moving a smaller aperture over this phase screen (see Fig. 2). In order to create a nonstationary turbulence, the wind speed varies by moving the aperture over the phase screen at a piecewise constant speed v_u. More specifically, we divide the simulation in a number of time sections of equal length, each of which has a constant wind speed in each section. In this simulation, we use the piecewise constant wind speed distribution in the horizontal direction: [4 1 3 9 5] pixels/sample with a simulation duration of $20 \cdot 10^3$ samples. Each piecewise constant section consists of $4 \cdot 10^3$ samples such that an oversized phase screen of size $(4 \cdot 10^3 \cdot (4+1+3+9+5) + m) \times m$ is generated.

A WFS consists of an $N \times N$ lenslet array with its lenslets evenly distributed on a square grid. Each of these lenslets measures the average slopes of the turbulence in both the horizontal and vertical direction, resulting in a total of $2N^2$ output channels. These slopes $Y(k) \in \mathbb{R}^{2N \times N}$ are related to the disturbed wavefront $\phi(k) \in \mathbb{R}^{m \times m}$ according to the following linear model:

$$\text{vec}(Y(k)) = G\text{vec}(\phi(k)) + w(k) \tag{13}$$

TABLE I
NUMERICAL SIMULATION PARAMETERS FOR SECTION IV-A

Model	
$N \times N$ WFS sensor points	9×9
SNR sensor noise	20 dB
D aperture diameter	1 m
Turbulence	
$m \times m$ turbulence phase screen	28×28
r_0 Fried parameter	0.2 m
L_0 outer scale	10 m
δ MA neighborhood	50
Horizontal wind speed	[4 1 3 9 5] pixels/sample
Vertical wind speed	0 pixels/sample
Identification data set	
N_t phase samples	10×10^3
N_a ALS iterations	20
Simulation data set	
N_t phase samples	20×10^3
λ forgetting parameter	0.9988
$P(1)$ initial value	I

where $G \in \mathbb{R}^{2N^2 \times m^2}$, and $w(k)$ is the sensor noise modeled as white Gaussian noise with a specific signal-to-noise ratio. To represent the open-loop system {turbulence + sensor}, a QUARKS model in open loop with temporal order 1 and Kronecker rank $r \geq 1$ is considered (3). In this brief, the covariance $E[v(k)v(k)^T]$ in (1) is not analyzed. Two data sets are generated under the same atmospheric conditions (see Table I). For the nonrecursive methods, one data set is used for offline identification and the other one is used for validation. The first data set is used to generate a starting value for the recursive identification methods.

The criterion stop for the ALS algorithm is reached when the difference of the residual in absolute value between two consecutive instants is less that 1×10^{-3} or a maximum of 20 N_a iterations is reached.

In order to illustrate the convergence of the recursive Kronecker algorithm, the variance accounted for (VAF) between the two signals $y(k)$ and $\hat{y}(k)$ is defined as

$$\text{VAF}(y(k), \hat{y}(k)) = \max\left(0, \left(1 - \frac{\frac{1}{2N^2}\sum_{i=1}^{2N^2} \|y_i(k) - \hat{y}_i(k)\|_2^2}{\frac{1}{2N^2}\sum_{i=1}^{2N^2} \|y_i(k)\|_2^2}\right) \times 100\right).$$

Since the turbulence is modeled as a stochastic process, we perform 100 Monte-Carlo simulations. In this experiment, the QUARKS algorithm in Algorithm 1 is used.

The number of iterations needed for ALS to achieve a difference in residual less than 1×10^{-3} is determined with and without normalization over 100 simulations. The average number of iterations needed with normalization is 8.53 and without normalization is 7.16. In Fig. 3, the accuracy of the estimates of the coefficient matrices over entire simulation duration is shown. The VAF is computed for each simulation. The mean and standard deviation over all Monte-Carlo simulations are then calculated and represented with the shaded area in Fig. 3. In red, the accuracy of the recursive QUARKS algorithm can be seen, compared with the accuracy of the nonrecursive QUARKS algorithm (blue) when the identification is performed on the whole data set assuming the latter is stationary.

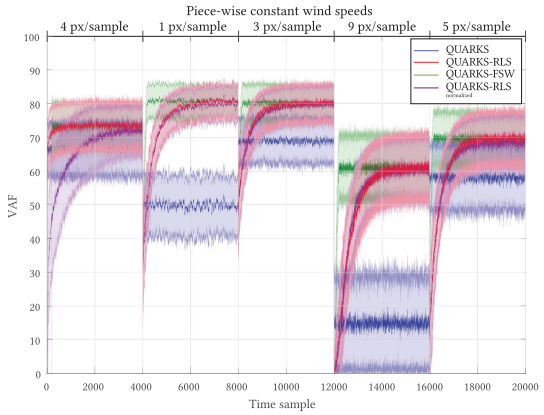

Fig. 3. Comparing the spatial VAF for QUARKS-RLS (red), FSW for QUARKS (green), and nonrecursive QUARKS model over the entire simulation duration assuming nonstationary turbulence (blue). The standard deviation for each of the three method is shaded.

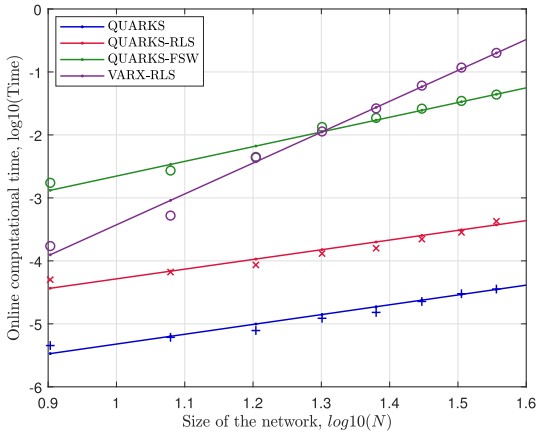

Fig. 4. Timings experiments for QUARKS (blue), recursive QUARKS (red), QUARKS with a FSW (green), and unstructured VARX-RLS (purple) for different sensor sizes. The coefficients of the models are: for QUARKS, $(a, b) = (1.56, -6.88)$; for QUARKS-RLS, $(a, b) = (1.54, -5.83)$; for QUARKS-FSW with length of sliding window 200, $(a, b) = (2.33, -4.99)$; and for VARX-RLS, $(a, b) = (4.90, -8.33)$.

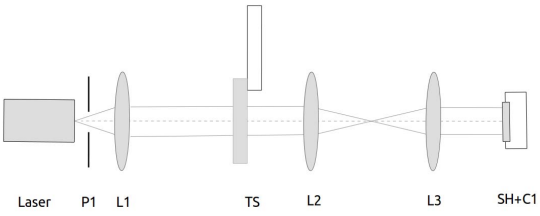

Fig. 5. Schematic of the laboratory testbed. The light emitted from the laser goes through the pupil P1 and the lens L1. It is collimated when reaching the turbulence plate TS that is placed at the focal plane of the lens L2. The lens L3 conjugates TS with the sensor SH + C1.

The recursive estimation with the scalable method proposed in Section II reaches higher performances when the temporal dynamics have reached stationarity. In green, the VAF is plotted for a QUARKS-fixed sliding window (FSW) model. This is obtained by estimating a QUARKS model at each new time sample, with the regression data within a FSW containing the last 200 time samples. This last method is an upper bound on the accuracy that can be achieved with the QUARKS method for nonstationary modeling at the expense of a much higher computational cost as a new ALS is solved at each time sample. The number of ALS iterations to meet the stopping criterion, however, is very small when the atmospheric conditions are slowly time varying. In purple, the recursive QUARKS algorithm is shown where normalization is applied to one of the factor matrices during recursive estimation. It shows that using normalization with recursive QUARKS decreases the rate of convergence and slightly decreases the overall accuracy of the algorithm.

We now investigate the *online* computational complexity in Fig. 4 with timing experiments on different sizes of the sensor. $2pr$ MMMs are required for QUARKS; Algorithm 3 and $2pr$ MMM are required for QUARKS-RLS; and Algorithm 1 and $2pr$ MMM are required for QUARKS using an FSW. A linear model $\log 10(\text{Time}) = a \times \log 10(N) + b$ was fit to the timing data, and we are particularly interested in the parameter a as it indicates how well the method scales with an increasing size of the sensor. The lower a is, the better the scalability will be. The online computational complexity for QUARKS, QUARKS-RLS, and QUARKS-FSW scale theoretically with $\mathcal{O}(N^3)$, and regression coefficients a of, respectively, 1.56, 1.54, and 2.33 are obtained. Although the size of the temporal window for QUARKS-FSW is constant over N, it still shows lower scalability than QUARKS and QUARKS-RLS. Furthermore, the regression coefficient for the unstructured RLS is 4.90 and hence a relative difference of 3.36 with QUARKS-RLS.

B. Laboratory Validation on a AO Testbed

We now consider the AO laboratory setup used to test the proposed identification approach. A schematic is shown in Fig. 5. The light is emitted from a He-Ne laser source ($\lambda = 635$ nm) and is then collimated into a beam of size $D = 9$ mm using the lens L1. The atmospheric turbulence for a single frozen layer is generated using a pseudorandom phase plate Turbulence Simulator (TS) machined by Lexitek Inc. The optical path difference is defined as follows. A phase design that follows the spatial Kolmogorov distribution is generated and is then multiplied by a factor that varies with an angle from the center of the array equal to $(1 + 1/5 \sin\theta)^{-5/6}$. The effect is to produce a phase design where the local value of the Fried parameter r_0 varies as $(1 + 1/5 \sin\theta)$. Different wind speed conditions are simulated by rotating the disk. In this brief, we propose to vary the speed to illustrate the improvements that the Kronecker-RLS algorithm enables over the batch method. Every 50 time samples, the speed in rounds per minute is set as follows:

$$\text{RPM} = \frac{1}{2} + \frac{T}{6} \sin\left(\frac{2\pi k}{N_t}\right) \tag{14}$$

where $T \in \{1, 1.5, 2, 2.5, 3\}$, and N_t is the total number of time samples.

The beam goes through the turbulence disk that is placed at the focal plane of the lens L2, $f_1 = 10$ cm. The lens L3 has a focal length of 10 cm and forms a telescope with L2. An OKOtech Shack–Hartmann WFS SH+C1, 1-in optical

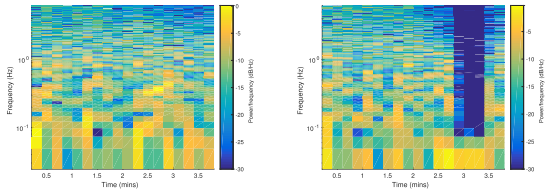

Fig. 6. Slope signal is considered in the middle of the aperture. Spectrogram for $T = 1$ (left). Spectrogram for $T = 3$ (right). The y-axis is in log10 scale. Each window contains 300 points and there is 50% overlap between the segments. The elements below -30 dB have been thresholded to 0 and appear in deep blue.

format, with a lenslet array pitch of 300 μm and focal length 18.6 mm, is placed perpendicular to the optical beam path at the focal point of L3. The turbulence phase profile and the grid of lenslets are on conjugated planes. An array of 28×28 lenslets is selected among which 566 are illuminated and considered as active. The Kronecker structure does not adapt well to circular apertures and we consider the rectangular aperture of the active lenslets. Therefore, the slope signals corresponding to the nonactive lenslets are set to 0, and hence, a total of $2 \times 28 \times 28$ sensors signals are considered. Such approximation implies larger prediction errors on the boundary of the pupil, although this effect is all the more mitigated if the factor matrices are sparse. At each time sample, the nonzero values predicted outside the circular aperture are set to 0.

The noise on the WFS affecting the slopes is a combination of different sources. The arrival of photons on the charge-coupled device (CCD) follows a Poisson distribution. The CCD has thermal and readout noises that are assumed independent and identically distributed. The finite discretization of the CCD plane with pixels also causes errors in computing the centroids. We assume nonetheless a zero-mean white noise and illustrate the robustness of the method with respect to this assumption.

The sampling frequency is $f_S = 12.5$ Hz. We introduce the Greenwood per sample frequency ratio, defined as

$$\frac{f_G}{f_S} = 0.427 \frac{v}{r_0} \frac{1}{f_S}.$$

It relates the wind speed with the sampling frequency and gives insight on how much the frozen flow has traveled over the aperture during a sample. It is upper bounded with 0.14, which is well below the Nyquist criteria, and hence, no temporal aliasing occurs.

We collect $N_t = 3 \times 10^3$ samples for each value of T. 0.5×10^3 samples are used for a batchwise identification of a QUARKS model, which serves as an initial guess for obtaining temporally varying estimates on the next 2.5×10^3 samples.

The temporal variations in the data are represented by the spectrogram using the short-time Fourier transform that illustrates the differences in the power spectrum of consecutive sliding windows [see Fig. 6]. Although the largest contribution is mostly located in the low frequencies, larger variations in the frequency content are observed for $T = 3$ than for $T = 1$.

The Kronecker rank and the temporal order both take values within the set $\{1, 3\}$. The accuracy is measured by calculating

TABLE II

LABORATORY TESTBED EXPERIMENT: VAF (%)

	QUARKS-RLS		QUARKS		Diag-RLS	Ratio
T, $r = 1$	$p = 1$	$p = 3$	$p = 1$	$p = 3$	$p = 1$	(%)
1	81.15	84.41	79.06	78.10	51.98	8.08
1.5	81.26	84.18	78.36	77.16	52.45	9.09
2	80.04	82.86	73.48	72.09	52.75	14.9
2.5	79.77	81.18	71.10	70.48	52.32	15.2
3	79.76	81.01	65.76	64.53	50.51	25.5
T, $r = 3$						
1	82.23	84.42	80.02	78.65		7.34
1.5	82.08	84.50	80.84	77.87		8.51
2	80.85	83.54	73.10	72.53		15.1
2.5	80.22	83.12	71.47	71.05		17.0
3	80.91	80.83	65.66	65.76		22.9

for each lenslet, the VAF (averaged in both the horizontal and vertical directions of the slopes signal) between the true signal Y and the reconstructed signal $\widehat{Y}$. Such measurement is different from Section IV-A in which the VAF was computed spatially for each time sample. We compare the accuracy of the QUARKS algorithm with the recursive QUARKS-RLS algorithm for varying conditions of nonstationarity in Table II. The relative improvement between the VAF for the QUARKS-RLS and the VAF for the QUARKS with $p = 3$ is indicated as *Ratio* in Table II.

In this AO configuration with one turbulence disk and at relatively low (f_G/f_S), increasing the temporal order or the Kronecker rank of the model leads to little improvements. Moreover, when increasing the amplitude T of the sine function, and hence the nonstationarity, the recursive algorithm is better equipped to handle the large changes induced by the varying rotational speed of the turbulence disk. The accuracy does decrease relatively less compared to the nonrecursive case.

VI. CONCLUSION

This brief introduces a scalable method for recursively identifying 2-D temporally varying systems. We have investigated in [9] the parameterization of the coefficient matrices as a sum of a few Kronecker products to decrease the computational requirements when identifying large-scale VAR models. Within the same class of structured matrices, we have analyzed the use of recursive algorithms for modeling nonstationary data. The estimates obtained with the QUARKS are used as initial guesses for the RLS steps that consist of sequentially updating the left and right factor matrices. It is also shown that the computational complexity of the QUARKS-RLS algorithm reduces from $\mathcal{O}(m^2n^2)$ to $\mathcal{O}(mn^2)$ compared to the unstructured VAR-RLS while guaranteeing low prediction error performances. Importantly, it alleviates the memory burden on QUARKS insofar as only the left and right covariance matrices along with the last measurement sample need to be stored, as opposed to the whole data set. A numerical validation for AO purposes was proposed on synthetic and laboratory testbed data. Although the discussion has dealt with a temporal order and a Kronecker rank equal to 1, the algorithms generalize as shown in the experimental section. The coefficient matrices A_i can be further decomposed into a product of many matrices $A_{1,i} \otimes \ldots \otimes A_{d,i}$ for d

larger than 2, which allows for further data compression. The QUARKS-RLS algorithm adapts by updating each factor matrix alternatively at each time sample.

VII. Additional Material

The code leading to these results is found online [17].

References

[1] F. Roddier, *Adaptive Optics in Astronomy*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[2] F. Roddier, "V The effects of atmospheric ttirhiilence in optical astronomy," *Prog. Opt.*, vol. 19, pp. 281–376, 1981.

[3] J. S. R. Gilmozzi and J. Spyromilio, "The European extremely large telescope (E-ELT)," *ESO Messenger*, vol. 127, pp. 11–19, 2007.

[4] K. Hinnen, M. Verhaegen, and N. Doelman, "A data-driven H_2-optimal control approach for adaptive optics," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 3, pp. 381–395, May 2008.

[5] O. Guyon and J. Males. (2017). "Adaptive optics predictive control with empirical orthogonal functions (EOFs)." [Online]. Available: https://arxiv.org/abs/1707.00570

[6] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[7] C. Correia, K. Jackson, J.-P. Véran, D. Andersen, O. Lardière, and C. Bradley, "Static and predictive tomographic reconstruction for wide-field multi-object adaptive optics systems," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 1, pp. 101–113, 2014.

[8] A. Chiuso and G. Pillonetto, "A Bayesian approach to sparse dynamic network identification," *Automatica*, vol. 48, no. 8, pp. 1553–1565, 2012.

[9] B. Sinquin and M. Verhaegen, "QUARKS: Identification of large-scale Kronecker vector-AutoRegressive models," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2018.2845662.

[10] C. F. Van Loan, "The ubiquitous Kronecker product," *J. Comput. Appl. Math.*, vol. 123, nos. 1–2, pp. 85–100, 2000.

[11] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica*, vol. 35, no. 7, pp. 1215–1241, 1999.

[12] A. H. Sayed and T. Kailath, "Recursive least-squares adaptive filters," in *The Digital Signal Processing Handbook*, vol. 21. Boca Raton, FL, USA: CRC Press, no. 1, 1998.

[13] S. S. Haykin, *Adaptive Filter Theory*. Delhi, India: Pearson Education, 2008.

[14] X. Wang, F. Ding, F. E. Alsaadi, and T. Hayat, "Convergence analysis of the hierarchical least squares algorithm for bilinear-in-parameter systems," *Circuits Syst. Signal Process.*, vol. 35, no. 12, pp. 4307–4330, 2016.

[15] A. Beghi, A. Cenedese, and A. Masiero, "Multiscale stochastic approach for phase screens synthesis," *Appl. Opt.*, vol. 50, no. 21, pp. 4124–4133, 2011.

[16] M. Le Ravalec, B. Noetinger, and L. Y. Hu, "The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning Gaussian simulations," *Math. Geol.*, vol. 32, no. 6, pp. 701–723, 2000.

[17] B. Sinquin, P. Varnai, G. Monchen, and M. Verhaegen. *Tensor Toolbox for Identifying Multi-Dimensional Systems*. [Online]. Available: https://bitbucket.org/csi-dcsc/t4sid