Preprints of the
15th IFAC Symposium on System Identification
Saint-Malo, France, July 6-8, 2009

Fast-array Recursive Closed-loop Subspace Model Identification ⋆

I. Houtzager * J.W. van Wingerden * M. Verhaegen *

* *Delft Center of Systems and Control, Delft University of Technology, Delft, 2628 CD The Netherlands (Tel: +3115 27 86707; e-mail: I.Houtzager@TUDelft.nl).*

Abstract: In this paper a subspace model identification algorithm is presented that can be implemented recursively to track slowly time-varying linear systems operating in open loop and closed loop. Particular attention is paid to the computational cost and tracking performance of the developed identification algorithm. The identification problem is described by only two linear problems. The computational complexity is reduced by using array algorithms to solve these linear problems and exploiting the structure in the vectors. This results in a fast implementation of the developed recursive identification algorithm. The effectiveness of the proposed algorithm in comparison with existing methods is emphasized with a simulation study on a time-varying closed-loop system.

Keywords: subspace identification, recursive identification, closed loop, tracking, time varying

1. INTRODUCTION

Subspace Model Identification (SMI) methods are efficient methods to identify Linear Time-Invariant (LTI) state-space models from Multi-Input and Multi-Output (MIMO) measurements of a dynamic system and are described in detail in Van Overschee and De Moor [1996], Verhaegen and Verdult [2007]. These methods store input and output data in structured block Hankel matrices, such that it is possible to retrieve certain subspaces that are related to the system matrices. These methods are very successful for offline identification, because the key linear algebra steps are a RQ factorization, an SVD, and the solution of a linear least-squares problem, therefore the problem of forming a nonlinear optimization is circumvented. Especially the use of a RQ factorization resulted in computationally efficient implementations of SMI schemes.

In online identification, it is important to update the identified model during the sampling period. In spite off the existence of fast updating algorithms for the RQ factorization, it is still difficult to implement these algorithms online due to the computational load of the SVD. Consequently, researchers try to find alternative algorithms for the SVD, or to avoid the application of the SVD, in order to apply the subspace concept in a recursive framework. The first successful Recursive Subspace Model Identification (RSMI) methods are described in Gustafsson [1998], Lovera et al. [2000], Lovera [2003] and are based on the Projection Approximation Subspace Tracking (PAST) method of Yang [1995]. More recently, new developments in the RSMI class of algorithms have been put forward by Mercère et al. [2003, 2008]. The proposed Propagator Method (PM) is based on the adaptation of a particular array signal processing technique to the RSMI problem. The main advantage of this approach over the previous concept is the use of a linear operator, which lead to recursive least-squares implementations of the algorithms.

Another disadvantage with the traditional SMI methods is that they give biased results when the system to be modelled operates in closed-loop, because the future inputs are correlated with the past noises, due to the feedback controller. Recently in Jansson [2003], Qin and Ljung [2003], Chiuso and Picci [2005], Chiuso [2007], a number of significant advances have been presented to identify an LTI state-space models from measurements of a dynamic system operating in closed loop. The Vector Auto Regressive with eXogenous inputs (VARX) models from the work of Chiuso [2007] have been used together with the PAST method in Wu et al. [2008] to create recursive implementations. VARX models, with high order, can provide asymptotical consistent estimates even on closed-loop data if there is sufficient excitation from an external signal or a controller of sufficiently high order. However, the estimation of VARX models still consumes a lot of computation time, therefore in the next sections of this paper a fast implementation of a novel closed-loop RSMI algorithm is proposed based on the optimized version of the Predictor-Based Subspace IDentification (PBSID) method, the so-called $\text{PBSID}_{\text{opt}}$ method in Chiuso [2007].

The remainder of this paper is as follows. In Section 2, the theoretical framework is presented for the subspace model identification of closed-loop LTI systems and the batch-wise solution is given. In Section 3, the fast implementation of the recursive solution for the closed-loop SMI problem is given and discussed. In Section 4, the effectiveness of the proposed RSMI algorithm in comparison with existing methods are emphasized with a simulation study on a time-varying system. In the final section we present the conclusions of this paper.

⋆ This research is supported under contract NO. TMR.5636 by the Dutch Technology Foundation STW.

2. SUBSPACE MODEL IDENTIFICATION FOR LTI SYSTEMS OPERATING IN CLOSED-LOOP

In this section a SMI method for LTI systems operating in closed-loop is presented that only requires the solution of linear problems. First, we describe a general problem formulation and second we explain the notations and assumptions made. In the third subsection, we introduce the data equation which will describe the main and largest linear problem to be solved. In the fourth subsection, the main estimation problem is solved batch-wise and it is shown how to obtain the system matrices. In the last subsection, a transformation is given which can be used as an alternative way to obtain the system matrices.

2.1 Problem formulation

Consider that the dynamics of the system to be modelled can be written in the following minimal state-space model in the innovation form:

$$\mathcal{S}\begin{cases} x_{k+1} &= Ax_k + Bu_k + Ke_k, \\ y_k &= Cx_k + Du_k + e_k, \end{cases} \tag{1}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^\ell$, are the state and output vectors, and $e_k \in \mathbb{R}^\ell$ denotes the zero-mean white innovation process noise. The state-space matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{\ell\times n}$, $D \in \mathbb{R}^{\ell\times r}$, and $K \in \mathbb{R}^{n\times \ell}$ are also called the system, input, output, direct feedthrough, and Kalman gain matrix, respectively. We can rewrite (1) in the predictor form as:

$$\begin{cases} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_k + Ky_k, \\ y_k &= Cx_k + Du_k + e_k, \end{cases} \tag{2}$$

with $\tilde{A} = A - KC$, and $\tilde{B} = B - KD$. It is well-known that an invertible linear transformation of the state does not change the input-output behaviour of a state-space system. Therefore, we can only determine the system matrices up to a similarity transformation $T \in \mathbb{R}^{n\times n}$: $T^{-1}AT$, $T^{-1}B$, $T^{-1}K$, CT, and D. The identification problem can now be formulated as:

Problem Description 1. Given the input sequence u_k, output sequence y_k over a time $k = \{0, \ldots, N-1\}$; find all, if they exist, system matrices A, B, C, D, and K up to a global similarity transformation both recursively and batch-wise.

2.2 Notation and assumptions

To make the notations more transparant, we define $m = r + \ell$, $z_k = \begin{bmatrix} u_k^T & y_k^T \end{bmatrix}^T$, $\bar{B} = \begin{bmatrix} \tilde{B} & K \end{bmatrix}$, and $\bar{D} = \begin{bmatrix} D & O^{\ell\times\ell} \end{bmatrix}$. Note that $O^{m\times n}$ is used to represent an m-by-n zero matrix; and I^m an m-by-m identity matrix. We define a past window denoted by $p \in \mathbb{N}^+$. This window is used to define the following stacked vector:

$$\bar{y}_k = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+p-1} \end{bmatrix}.$$

In a similar way we can obtain the stacked vectors $\bar{y}_{k-p}$, $\bar{e}_k$, $\bar{e}_{k-p}$, and $\bar{z}_{k-p}$.

The main assumptions are that the system to be modelled $\mathcal{S}$ is considered observable and that the noise sequence e_k needs to be white. Further, the problem formulation does not require any other assumptions on the correlation between the input and noise sequence, which opens the possibility to apply the algorithm in closed loop.

2.3 The data equation and the relation with the state

Before we present the well known data equation, we introduce the following matrices:

$$\tilde{\mathcal{H}} = \begin{bmatrix} \bar{D} & 0 & \cdots & 0 \\ C\bar{B} & \bar{D} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C\tilde{A}^{p-2}\bar{B} & C\tilde{A}^{p-3}\bar{B} & \cdots & \bar{D} \end{bmatrix}, \quad \tilde{\Gamma} = \begin{bmatrix} C \\ C\tilde{A} \\ \vdots \\ C\tilde{A}^{p-1} \end{bmatrix},$$

$$\tilde{\mathcal{K}} = \begin{bmatrix} \tilde{A}^{p-1}\bar{B} & \cdots & \tilde{A}\bar{B} & \bar{B} \end{bmatrix},$$

where $\tilde{\mathcal{H}} \in \mathbb{R}^{p\ell\times pm}$ is a lower block triangular Toeplitz matrix considering all $\bar{z}_k$ as inputs to the closed-loop observer system, $\tilde{\Gamma} \in \mathbb{R}^{p\ell\times n}$ is the extended observability matrix, and $\tilde{\mathcal{K}} \in \mathbb{R}^{n\times pm}$ is the extended controllability matrix. With these definitions the input-output behaviour of the model in (1) is now given by:

$$\bar{y}_k = \tilde{\Gamma}x_k + \tilde{H}\bar{z}_k + \bar{e}_k. \tag{3}$$

Now we are going to introduce in this procedure an approximation for the state. The state x_k is given by:

$$x_k = \tilde{A}^p x_{k-p} + \tilde{\mathcal{K}}\bar{z}_{k-p}, \tag{4}$$

where $\tilde{A}^p$ is the transition matrix. The main assumption in this section is that we assume that $\tilde{A}^j \approx 0$ for all $j \geq p$. It can be shown that if the system in (2) is uniformly exponential stable the approximation error can be made arbitrarily small by making p large, see Chiuso and Picci [2005], Chiuso [2007]. With this assumption the state x_k is approximately given by:

$$x_k \approx \tilde{\mathcal{K}}\bar{z}_{k-p}. \tag{5}$$

In a number of closed-loop SMI methods it is well known to make this approximation, see Jansson [2003], Chiuso [2007]. The output behaviour is now approximately given by:

$$y_k \approx C\tilde{\mathcal{K}}\bar{z}_{k-p} + Du_k + e_k. \tag{6}$$

With the approximation given in (5), we can rewrite (3) as:

$$\bar{y}_k \approx \widetilde{\Gamma\mathcal{K}}\bar{z}_{k-p} + \tilde{H}\bar{z}_k + \bar{e}_k. \tag{7}$$

The product $\widetilde{\Gamma\mathcal{K}}$ is now given by:

$$\widetilde{\Gamma\mathcal{K}} = \begin{bmatrix} C\tilde{A}^{p-1}\bar{B} & C\tilde{A}^{p-2}\bar{B} & \cdots & C\bar{B} \\ C\tilde{A}^{p}\bar{B} & C\tilde{A}^{p-1}\bar{B} & \ddots & C\tilde{A}\bar{B} \\ \vdots & \ddots & \ddots & \vdots \\ C\tilde{A}^{2p-1}\bar{B} & \cdots & C\tilde{A}^{p}\bar{B} & C\tilde{A}^{p-1}\bar{B} \end{bmatrix}. \tag{8}$$

With the assumption that $\tilde{A}^j \approx 0$ for all $j \geq p$, this expression can be approximated by the following upper block diagonal matrix:

$$\widetilde{\Gamma\mathcal{K}} \approx \begin{bmatrix} C\tilde{A}^{p-1}\bar{B} & C\tilde{A}^{p-2}\bar{B} & \cdots & C\bar{B} \\ 0 & C\tilde{A}^{p-1}\bar{B} & \ddots & C\tilde{A}\bar{B} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & C\tilde{A}^{p-1}\bar{B} \end{bmatrix}. \tag{9}$$

Due to the introduction of zeros in this matrix, the first block row in (9) can be used to construct the other block rows. Observe now that the product between the state and the observability matrix is approximately given by:

$$\bar{q}_k \triangleq \widetilde{\Gamma\mathcal{K}}\bar{z}_{k-p} \approx \tilde{\Gamma}x_k, \tag{10}$$

under the assumptions on the state it holds that:

$$\lim_{p\to\infty} \bar{q}_k \triangleq \widetilde{\Gamma\mathcal{K}}\bar{z}_{k-p} = \tilde{\Gamma}x_k, \tag{11}$$

and this observation is the key idea behind the $\text{PBSID}_{\text{opt}}$ method. This implies that we have to find an estimate of $C\tilde{\mathcal{K}}$ to construct $\widetilde{\Gamma\mathcal{K}}$. In (6), a linear problem is described in $C\tilde{\mathcal{K}}$ and consequently can be used to estimate $\widetilde{\Gamma\mathcal{K}}$ batch-wise or recursively. To summarize, after the construction of the matrix $\widetilde{\Gamma\mathcal{K}}$ we obtain a product between the observability and the state sequence. The approximation of the matrix $\widetilde{\Gamma\mathcal{K}}$ described in (9), which can be fully constructed by the product $C\tilde{\mathcal{K}}$ and is given by [1]:

$$\widetilde{\Gamma\mathcal{K}} \approx \begin{bmatrix} C\tilde{\mathcal{K}} \\ \left[O^{(r+\ell)\times(r+\ell)},\ C\tilde{\mathcal{K}}_{(:,1:(r+\ell)(p-1))}\right] \\ \left[O^{(r+\ell)\times 2(r+\ell)},\ C\tilde{\mathcal{K}}_{(:,1:(r+\ell)(p-2))}\right] \\ \vdots \\ \left[O^{(r+\ell)\times(r+\ell)(p-1)},\ C\tilde{\mathcal{K}}_{(:,1:(r+\ell))}\right] \end{bmatrix}. \tag{12}$$

2.4 Batch-wise solution

First we define the stacked vector Y:

$$Y = [y_{p+1}, \cdots, y_N],$$

In a similar way we can obtain the stacked vectors U, X. Further, we define the stacked matrix Z:

$$Z = [\bar{z}_1, \cdots, \bar{z}_{N-p+1}].$$

If the matrix $\Psi = \left[Z^T\ U^T\right]^T$ has full row rank, the matrices $C\tilde{\mathcal{K}}$ and D can be estimated by solving the following linear problem:

$$\min_{[C\tilde{\mathcal{K}}\ D]} \left\|Y - \left[C\tilde{\mathcal{K}}\ D\right]\Psi\right\|_F^2. \tag{13}$$

For finite p the solution of this linear problem will be biased due to the approximation made in (5). In the literature a number of papers appeared that studied the effect of the window size and although they proved the asymptotic properties of the algorithms (if $p \to \infty$ the bias disappears) it is hard to quantify the effect for finite p (Chiuso and Picci [2005], Chiuso [2007]).

The $\widetilde{\Gamma\mathcal{K}}Z$ is constructed using relation (12), which equals by definition the extended observability times the state sequence, $\tilde{\Gamma}X$. By computing a Singular Value Decomposition (SVD) of this estimate the state sequence and the order of the system is retrieved. Using the following SVD:

$$\widehat{\widetilde{\Gamma\mathcal{K}}Z} \approx [U\ U_\perp]\begin{bmatrix}\Sigma_n & 0 \\ 0 & \Sigma\end{bmatrix}\begin{bmatrix}V \\ V_\perp\end{bmatrix}, \tag{14}$$

where Σ_n is the diagonal matrix containing the n largest singular values and V is the corresponding row space. Note that we can find the largest singular values by detecting a gap between the singular values. The state sequence is now estimated by

$$\widehat{X} = \Sigma_n^{\frac{1}{2}}V. \tag{15}$$

It is well known that when the state, input, and output sequence are known, the system matrices A, B, C, and D can be estimated by solving the following linear problem[1]:

$$\min_\theta \left\| \begin{bmatrix} X_{(:,p+2:N)} \\ Y \end{bmatrix} - \underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\theta} \begin{bmatrix} X_{(:,p+1:N-1)} \\ U \end{bmatrix} \right\|_F^2. \tag{16}$$

In Verhaegen and Verdult [2007], it is also described to find a guarenteed stabilizing estimate of the observer matrix K using the Riccati equation.

2.5 Transformation of observability matrix

Although the calculation of the system matrices using the estimated state sequence is preferred, there is alternative way. Most recursive and traditional SMI methods obtain the system matrices A and C from the estimated observability matrix Γ as[1]:

$$A = \Gamma_{(1:(p-1)\ell,:)}^{\dagger}\Gamma_{(\ell+1:p\ell,:)}, \quad C = \Gamma_{(1:\ell,:)}.$$

The extended observability matrix Γ can be estimated by:

$$\widehat{\Gamma} = U\Sigma_n^{\frac{1}{2}}, \tag{17}$$

where U is the column space obtained from the SVD of the matrix $\Gamma\tilde{\mathcal{K}}Z$. In (15) the SVD of the matrix $\widetilde{\Gamma\mathcal{K}}Z$ is described instead, but fortunately there exist the following transformation:

$$\underbrace{\begin{bmatrix} \Gamma\tilde{\mathcal{K}}_{(0)} \\ \Gamma\tilde{\mathcal{K}}_{(1)} \\ \vdots \\ \Gamma\tilde{\mathcal{K}}_{(p-1)} \end{bmatrix}}_{\Gamma\tilde{\mathcal{K}}} = \underbrace{\begin{bmatrix} I^\ell & 0 & \cdots & 0 \\ CK & I^\ell & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{p-2}K & CA^{p-1}K & \cdots & I^\ell \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} \widetilde{\Gamma\mathcal{K}}_{(0)} \\ \widetilde{\Gamma\mathcal{K}}_{(1)} \\ \vdots \\ \widetilde{\Gamma\mathcal{K}}_{(p-1)} \end{bmatrix}}_{\widetilde{\Gamma\mathcal{K}}}, \tag{18}$$

which brings the prediction observability matrix to the following innovation form:

$$\Gamma\tilde{\mathcal{K}} = \begin{bmatrix} C\tilde{A}^{p-1}\bar{B} & C\tilde{A}^{p-2}\bar{B} & \cdots & C\bar{B} \\ * & CA\tilde{A}^{p-2}\bar{B} & \ddots & CA\bar{B} \\ \vdots & \ddots & \ddots & \vdots \\ * & \cdots & * & CA^{p-1}\bar{B} \end{bmatrix}, \tag{19}$$

where $\widetilde{\Gamma\mathcal{K}}_{(i)}$ denotes the ith block row of $\widetilde{\Gamma\mathcal{K}}$, and $*$ denotes non-essential values which can be made zero. The Markov parameters of the impulse matrix W are not known in advance, but in Dong et al. [2008], de Korte [2009] a similar transformation is described as follows[1]:

$$\Gamma\tilde{\mathcal{K}}_{(i)} = \widetilde{\Gamma\mathcal{K}}_{(i)} + \sum_{j=0}^{i-1}\Big(\underbrace{C\tilde{A}^{i-j-1}K}_{C\tilde{\mathcal{K}}_{(:,(p-i+j)(r+\ell)+r+(1:\ell))}}\ \Gamma\tilde{\mathcal{K}}_{(i)}\Big), \tag{20}$$

where all the Markov parameters are known. Then, by applying the SVD on the matrix $\Gamma\tilde{\mathcal{K}}$ instead, an estimate of the matrix Γ can be found in the column space. Given the estimates of A and C, the system matrix B can be obtained (D can be obtained from (6)) from another linear problem, see for batch-wise Van Overschee and De Moor [1996], Verhaegen and Verdult [2007] and recursively Lovera et al. [2000]. Further, it is noted that applying this transformation can also lead to a weighted version of $\text{PBSID}_{\text{opt}}$, see Chiuso [2007].

[1] For simplicity MATLAB notation is used.

3. RECURSIVE SOLUTION OF THE CLOSED-LOOP SUBSPACE MODEL IDENTIFICATION METHOD

In this section the recursive solution of the presented closed-loop SMI method is presented. With the batch-wise solution the order of the system was unknown, and could be obtained from the singular values. For the recursive version of the algorithm the number of states n is assumed to be known. In the first subsection, the recursive solution is described in three steps. In the second and last subsection, we look at the implementation and the computational cost of the recursive least-squares solvers.

3.1 Recursive identification steps

The recursive solution of the presented closed-loop SMI scheme is given in Table 1 and is divided in the following three steps:

Step 1: The estimation of $C\tilde{\mathcal{K}}$ and construction of $\bar{q}_k$ The linear problem formulated in (13) derived from the relation in (5) can be written as a recursive least-squares problem, see Section 3.2. A very fast solution of this problem can be computed recursively with the fast-array scheme given in Table 1. From the recursive estimate of $C\tilde{\mathcal{K}}$ the propagator vector $\bar{q}_k$ can be constructed.

Step 2: The estimation of x_k The state x_k can be constructed from the recursive estimate of $\widetilde{\Gamma\mathcal{K}}$. To construct the state at time instance x_k in the same state basis as x_{k-1}, the Propagator Method (PM), described in Mercère et al. [2008], can be used. Assuming that the system (1) is observable, then $\tilde{\Gamma}$ has at least n linearly independent rows. If the order n is known, it is possible to build a permutation matrix $S \in \mathbb{R}^{\ell p\times \ell p}$ such that the extended observability matrix can be decomposed in the following way:

$$\begin{bmatrix}\bar{q}_{k,1}\\ \bar{q}_{k,2}\end{bmatrix} \triangleq S\widetilde{\Gamma\mathcal{K}}\bar{z}_{k-p} = \begin{bmatrix}\tilde{\Gamma}_1\\ \tilde{\Gamma}_2\end{bmatrix}\tilde{\mathcal{K}}\bar{z}_{k-p} = \begin{bmatrix}I^n\\ P\end{bmatrix}\tilde{\Gamma}_1\tilde{\mathcal{K}}\bar{z}_{k-p}, \quad (21)$$

where $\tilde{\Gamma}_1$ are the blocks of n independent rows and $\tilde{\Gamma}_2$ are the matrices of the $\ell p - n$ others, and P is a unique operator named the propagator. This relation and the approximation in (10) implies that an estimate of the state can be calculated in a particular basis, defined by:

$$\hat{x}_k = \bar{q}_{k,1} \approx \tilde{\Gamma}_1 x_k. \quad (22)$$

How to find the permutation matrix S, without knowing $\tilde{\Gamma}$, such that the first n rows of $S\tilde{\Gamma}$ are linearly independent is discussed in Mercère et al. [2008]. The PM method is also useful for recursive implementation of the alternative solution mentioned in Section 2.5.

Step 3: The recursive estimation of the system matrices From the estimate of the state update, the system matrices can be updated using recursive versions of the linear least-squares problem (16).

3.2 Recursive least squares solvers

Let λ be a positive scalar, usually very close to one, say $0 \ll \lambda \leq 1$. The solution θ can be computed recursively, and the cost in (13) and (16) is replaced by:

$$\min_{\theta} \sum_{k=0}^{N} \lambda^{N-k} \|d_k - \theta w_k\|_F^2 . \quad (23)$$

The scalar λ is called the forgetting factor since past data are exponentially weighted less heavily than more recent data. It is common in recursive identification to employ an exponentially weighted regularized least-squares cost function, because its purpose is to give more weight to recent data and less weight to data from the remote past, such that time-varing tracking of θ becomes possible.

A good overview of RLS schemes, which minimizes the cost given in (23), is found in the book of Sayed [2008]. The conventional RLS implementations are used in a broad class of applications, such as RSMI, because of the simplicity of these schemes. However, in the current decade the tendency has arisen to exclude the implementation referred to in Verhaegen [1989] as the conventional RLS implementation for practical use and opts the use of a square-root type of implementation due to better numerical robustness against round-off errors. The array methods written in Table 1 are powerful variants of square-root RLS that performs the computations in a reliable manner using a sequence of elementary Givens and/or hyperbolic rotations, see for stable implementations Chandrasekaran and Sayed [1996]. Another advantage is that array methods can be used to exploit any shifting structure in the data. Using a fast-array RLS scheme that exploits the shifting structure of the data vector $\bar{z}_{k-p}$ in (7) and results in an algorithm with computational load of $\mathcal{O}(p)$, instead of $\mathcal{O}(p^2)$, see Table 1.

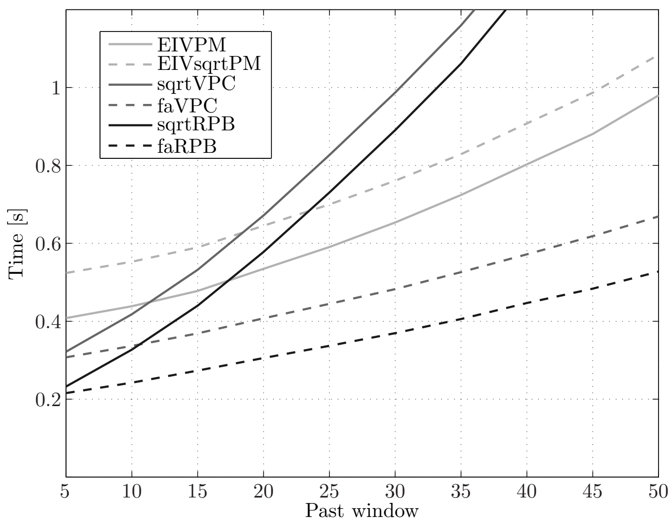

Fig. 1. Computational time of EIVPM, EIVsqrtPM, sqrtVPC, faVPC, sqrtRPB, and faRPB for different past (future) window sizes. The following parameters have been used: $N = 1000$, MCS = 100.

A comparison of the computational time needed for six RSMI schemes to compute the system matrices from random measurement data with 1000 samples and different past (and future) window sizes is illustrated in Fig. 1. The computational time are averaged over 100 Monte Carlo Simulations (MCS). The proposed RSMI method is denoted by sqrtRPB for the square-root RLS implementation and faRPB for fast-array RLS implementation. The methods EIVPM and EIVsqrtPM (square root version

Table 1. The fast-array closed-loop RSMI algorithm and its computational load of $\mathcal{O}\left((r+\ell)^2 p\right)$

init

Given θ_p, λ_1, λ_2, L_p, P_p, and S.

for $k = p+1, p+2, \ldots$

Step 1:

Define

$$\bar{w}_k = \begin{bmatrix} \bar{w}_{k-1(r+\ell+1:(r+\ell)p,:)} \\ y_{k-1} \\ u_k \end{bmatrix}$$

Find a J-unitary matrix Θ_1 , where $J = \operatorname{diag}\left(1, -I^{r+\ell}, I^{r+\ell}\right)$, such that

$$\begin{bmatrix} \gamma_{1,k-1}^{-1/2} & \frac{1}{\sqrt{\lambda_1}}\bar{w}_k^T L_{k-1} \\ \begin{bmatrix} g_{1,k-1}\gamma_{1,k-1}^{-1/2} \\ 0 \end{bmatrix} & \frac{1}{\sqrt{\lambda_1}} L_{k-1} \end{bmatrix} \Theta_{1,k} = \begin{bmatrix} \gamma_{1,k}^{-1/2} & \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ g_{1,k}\gamma_{1,k}^{-1/2} \end{bmatrix} & L_k \end{bmatrix}, \qquad 14\,(r+\ell)^2\,(p+1) - 6\,(r+\ell)$$

$$\begin{bmatrix} C\tilde{\mathcal{K}}_k & D_k \end{bmatrix} = \begin{bmatrix} C\tilde{\mathcal{K}}_{k-1} & D_{k-1} \end{bmatrix} + \left(y_k - \begin{bmatrix} C\tilde{\mathcal{K}}_{k-1} & D_{k-1} \end{bmatrix}\bar{w}_k\right)\left(g_{1,k}\gamma_{1,k}^{-1/2}\gamma_{1,k}^{1/2}\right)^T, \qquad (4\ell+1)\,(r+\ell)\,(p+1)$$

$$\widetilde{\Gamma\mathcal{K}}_k = \begin{bmatrix} C\tilde{\mathcal{K}}_{k(:,1:(r+\ell)p)} \\ \begin{bmatrix} O^{(r+\ell)\times(r+\ell)}, & C\tilde{\mathcal{K}}_{k(:,1:(r+\ell)(p-1))} \end{bmatrix} \\ \begin{bmatrix} O^{(r+\ell)\times 2(r+\ell)}, & C\tilde{\mathcal{K}}_{k(:,1:(r+\ell)(p-2))} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} O^{(r+\ell)\times(r+\ell)(p-1)}, & C\tilde{\mathcal{K}}_{k(:,1:(r+\ell))} \end{bmatrix} \end{bmatrix},$$

Step 2:

$$\bar{q}_k = S\widetilde{\Gamma\mathcal{K}}_k \bar{z}_{k-p}, \qquad (2\ell p - 1)\,(n+r)$$

$$x_k = \bar{q}_{k(1:n,:)},$$

Step 3:

Find a unitary matrix Θ_2 such that

$$\begin{bmatrix} I & \frac{1}{\sqrt{\lambda_2}}\begin{bmatrix} x_{k-1} \\ u_{k-1} \end{bmatrix}^T P_{k-1}^{1/2} \\ 0 & \frac{1}{\sqrt{\lambda_2}} P_{k-1}^{1/2} \end{bmatrix} \Theta_{2,k} = \begin{bmatrix} \gamma_{2,k}^{-1/2} & 0 \\ g_{2,k}\gamma_{2,k}^{-1/2} & P_k^{1/2} \end{bmatrix}, \qquad 10\,(n+r)^2 + 2\,(n+r)$$

$$\theta_k = \theta_{k-1} + \left(\begin{bmatrix} x_k \\ y_{k-1} \end{bmatrix} - \theta_{k-1}\begin{bmatrix} x_{k-1} \\ u_{k-1} \end{bmatrix}\right)\left(g_{2,k}\gamma_{2,k}^{-1/2}\gamma_{2,k}^{1/2}\right)^T, \qquad (4\,(n+\ell)+1)\,(n+r)$$

end

of the previous) are found in Mercère et al. [2008], and the methods sqrtVPC and faVPC are square-root and fast-array RLS implementations of the closed-loop RSMI method found in Wu et al. [2008]. From the figure, it is clearly visible that the proposed RPB method has the fastest computation time, even for large past window sizes if fast-array RLS is used.

4. SIMULATION STUDY

In order to illustrate the performances of the RSMI methods proposed in this paper and the existing methods, the state-space system from Mercère et al. [2008] is used:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.3 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ 0 & -0.6 \\ 0.5 & 0 \end{bmatrix} u_k \\ &\quad + \begin{bmatrix} 0.055 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.045 \end{bmatrix} w_k, \\ y_k &= \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.025 & 0 \\ 0 & 0.03 \end{bmatrix} v_k, \end{aligned} \qquad (24)$$

where v_k and w_k are random white noise sequences of variance 1. To create a closed-loop system, a time-varying state-feedback control law is applied as described in Ogata [1994] that stabilizes the above system over the whole trajectory. Two practical situations are considered to evaluate the tracking performance: a slowly time-varying case, and an abrupt-change case.

4.1 Slowly time-varying case

After a time-invariant phase of 665 samples, the following state matrix is used:

$$A + \operatorname{diag}(\,-0.3,\ -0.5,\ 0.2\,)\frac{\exp\left(-(k-665)/2000\right) - 1}{\exp\left(-1\right) - 1}. \qquad (25)$$

Thus the poles of the systen drift from $\{0.3, 0.5, 0.8\}$ during the next 1335 samples. The estimated poles trajectories averaged over the 100 MCS are displayed in Fig. 2. As expected the open-loop EIVPM method gives biased results, because the system to be modelled operates in closed-loop. The EIVPM method cannot handle the problem that the future inputs are correlated with past noises. The VPC and the presented RPB method follow the eigenvalue trajectories much better. From the averaged responses, it visible that the VPC algorithm has more difficulties and sometimes give some biased results. Further, it was observed from the simulations that the variance around the mean trajectories is much larger for the VPC method compared to the RPB method.

4.2 Abrupt-change case

After a time-invarient phase of size 665 samples, the following abrupt change $A(3,3) = 0.65$ in the state matrix is applied. Thus the pole 0.5 shifts to 0.65. The estimated poles trajectories averaged over the 100 MCS are displayed in Fig. 3. Similar as in Section 4.1, the open-loop EIVPM method gives biased results. In the figure, the method VPC shows a faster convergence in spite of some bias. Also here, it was shown in simulation that the individual responses of RPB are much smoother.

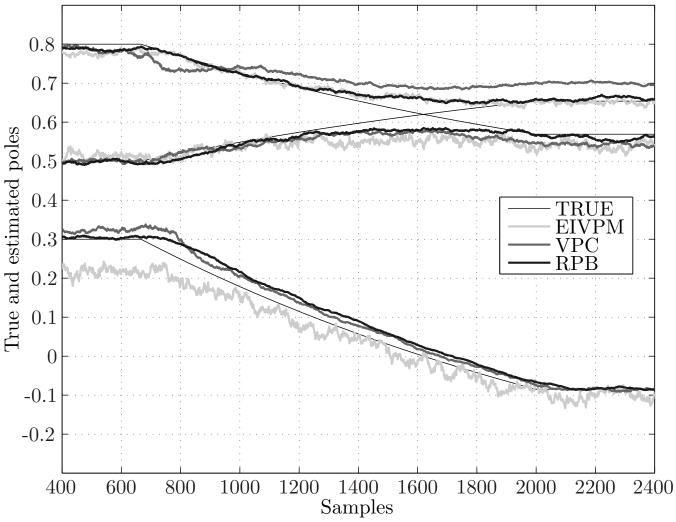

Fig. 2. Trajectories of the estimated poles using EIVPM, VPC, and RPB in a slowly changing environment. The following parameters are used: $\lambda_1 = \lambda_2 = 0.98$, $p = 5$, SNR = 25dB, and MCS = 100.

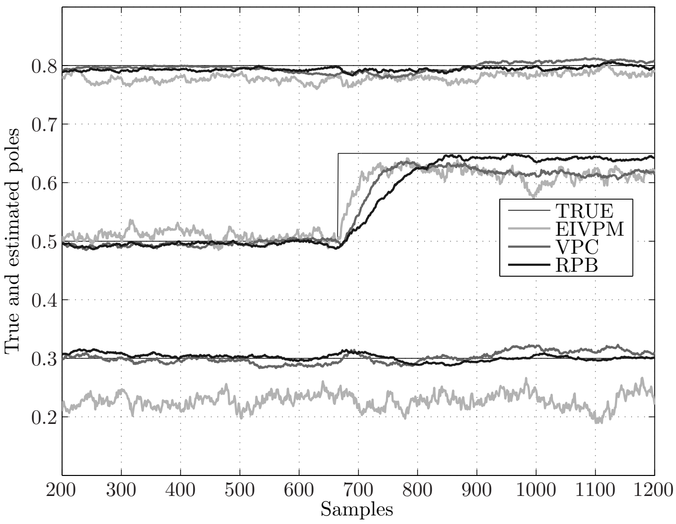

Fig. 3. Trajectories of the estimated poles using EIVPM, VPC, and RPB in an abrupt time-varying environment. The following parameters are used: $\lambda_1 = \lambda_2 = 0.98$, $p = 5$, SNR = 25dB, and MCS = 100.

5. CONCLUSION

A subspace model identification algorithm is presented that can be implemented recursively to track time-varying linear systems operating in open loop and closed loop. The identification problem is described by only two linear problems. The computational complexity is reduced by using array algorithms to solve these linear problems and exploiting the structure of the vectors. This results in a fast implementation of the developed recursive identification algorithm. The effectiveness of the proposed algorithm in comparison with existing methods is emphasized with a simulation study on a time-varying closed-loop system.

REFERENCES

S. Chandrasekaran and A. H. Sayed. Stabilizing the generalized schur algorithm. *SIAM Journal on Matrix Analysis & Applications*, 17(4):950–983, 1996.

A. Chiuso. The role of vector auto regressive modeling in predictor-based subspace identification. *Automatica*, 43 (6):1034–1048, 2007.

A. Chiuso and G. Picci. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.

R.B.C. de Korte. Subspace-based identification techniques for a 'smart' wind turbine rotor blade. Master's thesis, Delft University of Technology, The Netherlands, 2009.

J. Dong, M. Verhaegen, and E. Holweg. Closed-loop subspace predictive control for fault tolerant MPC design. In *17th IFAC World Congress*, Seoul, Korea, 2008.

T. Gustafsson. Instrumental variable subspace tracking using projection approximation. *IEEE Transactions on Signal Processing*, 46:669–681, 1998.

M. Jansson. Subspace identification and ARX modeling. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

M. Lovera. Recursive subspace identification based on projector tracking. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

M. Lovera, Gustafsson T., and M. Verhaegen. Recursive subspace identification of linear and non-linear wiener state-space models. *Automatica*, 36(11):1639–1650, 2000.

G. Mercère, S. Lecœuche, and C. Vasseur. A new recursive method for subspace identification of noise systems: EIVPM. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

G. Mercère, L. Bako, and S. Lecœuche. Propagator-based methods for recursive subspace model identification. *Signal Processing*, 88(3):468–491, 2008.

K. Ogata. *Discrete-Time Control Systems*. Prentice hall, 1994.

S.J. Qin and L. Ljung. Closed-loop subspace identification with innovation estimation. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

A. H. Sayed. *Adaptive Filters*. John Wiley & Sons, 2008.

P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems; Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.

M. Verhaegen. Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes. *Automatica*, 25(3):437–444, 1989.

M. Verhaegen and V. Verdult. *Filtering and System Identification*. Cambridge University Press, 2007.

P. Wu, C. Yang, and Z. Song. Recursive subspace model identification based on vector autoregressive modelling. In *17th IFAC World Congress*, Seoul, Korea, 2008.

B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.