Propagator-based methods for recursive subspace model identification[*]

Guillaume Mercère
Université de Poitiers
Laboratoire d'Automatique et d'Informatique Industrielle
40 avenue du recteur Pineau
86022 Poitiers, France
Email: guillaume.mercere@univ-poitiers.fr

Laurent Bako
École des Mines de Douai
Département Informatique et Automatique
941, rue Charles Bourseul B.P. 838
59508 Douai Cedex, France
Email: bako@ensm-douai.fr

Stéphane Lecœuche
École des Mines de Douai
Département Informatique et Automatique
941, rue Charles Bourseul B.P. 838
59508 Douai Cedex, France
Email: lecoeuche@ensm-douai.fr

Abstract

The problem of the online identification of multi-input multi-output (MIMO) state-space models in the framework of discrete-time subspace methods is considered in this paper. Several algorithms, based on a recursive formulation of the MIMO output error state-space (MOESP) identification class, are developed. The main goals of the proposed methods are to circumvent the huge complexity of eigenvalues or singular values decomposition techniques used by the offline algorithm and to provide consistent state-space matrices estimates in a noisy framework. The underlying principle consists in using the relationship between array signal processing and subspace identification to adjust the propagator method (originally developed in array signal processing) to track the subspace spanned by the observability matrix. The problem of the (coloured) disturbances acting on the system is solved by introducing an instrumental variable in the minimised cost functions. A particular attention is paid to the algorithmic development and to the computational cost. The benefits of these algorithms in comparison with existing methods are emphasised with a simulation study in time-invariant and time-varying scenarios. matrices.

Key words: System identification, recursive algorithm, subspace method, instrumental variable, multivariable system

[*] Published in Signal Processing, Vol. 88, pp. 468–491, 2008

1 Introduction

Subspace-based approach for system identification is a particular field of experimental modelling which has reached a definite maturity from now on, as shown by numerous reference articles [16, 38, 36, 43, 2] and various application papers [42, 15, 1, 26, 7]. Offline subspace identification methods are indeed attractive since a state-space realization can be directly estimated from input/output (I/O) data without non linear optimization (generally required by the prediction methods [17]). Furthermore, these techniques are characterised by the use of robust numerical tools such as the RQ factorization and the singular values decomposition (SVD). Interesting from a numerical point of view, the batch subspace model identification (SMI) algorithms are not usable for online implementation because of the SVD computational complexity. Indeed, in many online identification scenarios, it is important to update the model as time goes on with a reduced computational cost. Consequently, it was necessary to find SVD alternative algorithms in order to apply the subspace concept in a recursive framework. First, some works proposed adaptations of SMI methods in order to update the SVD [39, 4]. Unfortunately, these techniques had the drawback of requiring that the disturbances acting on the system outputs were spatially and temporally white, which is obviously restrictive in practice. Then, recursive subspace model identification (RSMI) methods based on the Yang's criterion and the projection approximation subspace tracking (PAST) cost function [44] were introduced. These specific techniques were developed to track the subspace spanned by the extended observability matrix in a coloured disturbances framework [11, 20, 30, 19]. More precisely, instrumental variable adaptations of the PAST technique to the problem of RSMI were considered, the focus being computational efficiency [11] or estimation accuracy [20, 30]. More recently, in order to provide solutions to the approximation induced by the reduction of the fourth order Yang's criterion to a quadratic cost function, new developments in the RSMI class of algorithms have been put forward [23, 21, 24]. The proposed methods are based on the adaptation of a particular array signal processing technique to the recursive subspace identification problem: the propagator [27]. The main advantage of this approach over the previous conception lies in the use of a linear operator and quadratic criteria which lead to recursive least squares implementations for the algorithms. This characteristic has newly allowed the analysis of the convergence properties of the developed techniques for the recursive update of the subspace estimates [25]. It has also been used with success to online track the modal parameters of airplanes during test flights [6].

As shown in the following, all the propagator-based techniques are made up of two stages:

1. the online update of a particular vector named the observation vector,
2. the recursive estimation of the extended observability matrix from the online update of the propagator.

An extensive literature is already dedicated to the problem of the observation vector estimation [11, 20, 30, 21, 12]. Therefore, only a brief overview of the main approaches will be introduced in the following. On the other hand, new results concerning the observability matrix estimation phase have been proposed since 2003 [22, 19, 21, 24], mostly based on the propagator [27]. Thus, the main goal of this paper consists in introducing these recent recursive propagator-based subspace methods in a unified framework. More precisely, several (new) algorithms, based on particular unconstrained and quadratic cost functions, are introduced and compared in a performance and numerical complexity point of view.

It is important to stress on the fact that most of the methods used for subspace tracking in array signal processing [3, 32, 44] assume that the noise acting on the antenna is white. The propagator method [27], which will be adapted in the following, does not depart from this condition. Now, in system identification, this hypothesis is seldom satisfied in practice. The problem of the state-space matrices estimation in the presence of output measurement disturbances and process noise will be thereby analysed. More precisely, to provide unbiased estimates in a general noisy framework, instrumental variables will be used either in the online observation vector estimation step or in the recursive propagator updating phase. A specific attention will be paid to the computational cost of these two alternatives.

The outline of this paper is as follows: in Section 2, the system model is introduced, the main notations are defined, the general assumptions are stated and a short description of the analogy between recursive subspace identification and subspace tracking in array signal processing is proposed. This analogy allows to display both steps composing the developed recursive subspace identification approach. Section 3 is dedicated to the online estimation of the observation vector from the new I/O data thanks to the use of some specific RQ factorization updates. The problem of the recursive computation of the propagator and, by extension of a basis of the observability subspace, is studied in Section 4. The adaptation of the propagator method in the recursive identification framework is more precisely considered. The necessity to apply a particular permutation matrix and instrumental variables to consistently and recursively identify MIMO systems in the presence of disturbances is also analysed. This study leads to the development of several algorithms, the numerical complexity of which is examined. In Section 5, the performances of these new techniques are emphasised from numerical simulations. Section 6 concludes the paper.

2 Notations and problem formulation

Consider the following finite dimensional causal linear time-invariant state-space model

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \tag{1a}$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{v}(t) \tag{1b}$$

with n_u inputs $\mathbf{u}$, n_y outputs $\mathbf{y}$, a n_x dimensional state vector $\mathbf{x}$, a process noise sequence $\mathbf{w}$ and output measurement disturbances $\mathbf{v}$. Assume furthermore that

- the system is asymptotically stable and observable,
- there is no feedback from $\mathbf{y}$ to $\mathbf{u}$,
- both perturbation vectors are stationary zero mean white Gaussian noise such that

$$\mathbb{E}\left\{\begin{bmatrix}\mathbf{w}(i)\\ \mathbf{v}(i)\end{bmatrix}\begin{bmatrix}\mathbf{w}^T(j) & \mathbf{v}^T(j)\end{bmatrix}\right\} = \begin{bmatrix}\mathbf{R}_{\mathbf{w}} & \mathbf{R}_{\mathbf{wv}}\\ \mathbf{R}_{\mathbf{vw}} & \mathbf{R}_{\mathbf{v}}\end{bmatrix}\delta(i,j) \tag{2}$$

  where $\delta(i,j)$ denotes the Kronecker delta function and $\mathbb{E}\{\bullet\}$ the mathematical expectation,
- $\mathbf{v}$ and $\mathbf{w}$ are independent of the initial state $\mathbf{x}(0)$.

In the following, with some abuse of notations, the cross-covariance matrix will be noted as $\mathbf{R}_{\mathbf{ab}} = \mathbb{E}\left\{\mathbf{a}(k)\mathbf{b}^T(k)\right\}$and estimates of signal correlations will be denoted by $\hat{\mathbf{R}}_{\mathbf{ab}}(t) = \sum_{k=1}^{t}\beta^{t-k}\mathbf{a}(k)\mathbf{b}^T(k)$ where $0 < \beta \leq 1$ is a forgetting factor.

The goal of the RSMI methods consists in online estimating the system matrices $[\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}]$ at each new I/O data acquisition. The algorithms developed in this paper are based on the MIMO Output Error State-sPace (MOESP) class approach [40, 41, 37, 38]. The key problem of these subspace identification schemes is the consistent estimation of the extended observability matrix column space defined as (f is a user fixed integer chosen such that $f > n_x$ [13])

$$\mathbf{\Gamma}_f = \begin{bmatrix}\mathbf{C}^T & (\mathbf{CA})^T & \cdots & (\mathbf{CA}^{f-1})^T\end{bmatrix}^T$$

from measured I/O samples. It is indeed relatively straightforward to extract the state-space matrices (up to a similarity transformation) from $\mathbf{\Gamma}_f$ by exploiting particular properties of this matrix such that its shift invariance (see *e.g.* [13, Chapter 6] for an overview of the the state-space matrices estimation techniques). The starting point of the MOESP schemes for the estimation of the column space of $\mathbf{\Gamma}_f$ is the so called "data equation" [38]

$$\mathbf{Y}_f(\tau) = \mathbf{\Gamma}_f\mathbf{X}(\tau) + \mathbf{H}_f\mathbf{U}_f(\tau) + \underbrace{\mathbf{G}_f\mathbf{W}_f(\tau) + \mathbf{V}_f(\tau)}_{\mathbf{B}_f(\tau)}$$

where $\mathbf{U}_f$, $\mathbf{Y}_f$, $\mathbf{W}_f$ and $\mathbf{V}_f$ are Hankel matrices defined as follows

$$\mathbf{U}_f(\tau) = \begin{bmatrix}\mathbf{u}(t) & \cdots & \mathbf{u}(t+M-1)\\ \mathbf{u}(t+1) & \cdots & \mathbf{u}(t+M)\\ \vdots & \ddots & \vdots\\ \mathbf{u}(t+f-1) & \cdots & \mathbf{u}(t+f+M-2)\end{bmatrix} \text{ with } \begin{cases}M >> f > n_x\\ \tau = t+M-1,\end{cases}$$

where $\mathbf{H}_f$ and $\mathbf{G}_f$ are the block Toeplitz matrices of the impulse response respectively from $\mathbf{u}$ to $\mathbf{y}$ and from $\mathbf{w}$ to $\mathbf{y}$

$$\mathbf{H}_f = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \cdots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{f-2}\mathbf{B} & \mathbf{CA}^{f-3}\mathbf{B} & \cdots & \mathbf{D} \end{bmatrix}, \quad \mathbf{G}_f = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CA} & \mathbf{C} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{f-2} & \mathbf{CA}^{f-3} & \cdots & \mathbf{0} \end{bmatrix}$$

and where $\mathbf{X} = \begin{bmatrix}\mathbf{x}(t) & \cdots & \mathbf{x}(t+M-1)\end{bmatrix}$. On the basis of this data equation, it is easy to show that

$$\mathrm{span}_{\mathrm{col}}\left\{\mathbf{Y}_f\right\} \subseteq \mathrm{span}_{\mathrm{col}}\left\{\mathbf{\Gamma}_f\right\} + \mathrm{span}_{\mathrm{col}}\left\{\mathbf{H}_f\right\} + \mathrm{span}_{\mathrm{col}}\left\{\mathbf{B}_f\right\}. \tag{3}$$

The identification problem is then to isolate $\mathrm{span}_{\mathrm{col}}\left\{\mathbf{\Gamma}_f\right\}$ from the knowledge of $\mathrm{span}_{\mathrm{col}}\left\{\mathbf{Y}_f\right\}$. A number of versions of MOESP were developed in order to work out accurate estimates of the column space of the observability matrix from the available I/O data. The algorithms mainly differ according to the assumptions on the disturbances acting on the system generating the data [38, 5].

The problem studied in this paper is the recursive extraction of $\mathrm{span}_{\mathrm{col}}\left\{\mathbf{\Gamma}_f\right\}$. In order to avoid the use of classic burdensome tools such as the SVD, it is proposed to adapt particular SVD alternatives, initially used in array signal processing [14]. This approach can be implicitly justified by the fact that the mathematical problem in both fields is the same: to track some eigencomponents of particular matrices by adapting specific subspaces with the last observations. More precisely, in subspace tracking in array signal processing [14], the considered problem consists in recursively determining the directions of arrival $\boldsymbol{\theta}$ by online estimating the column subspace of the steering matrix $\mathbf{\Gamma}(\boldsymbol{\theta})$ from the following data generation model

$$\mathbf{z}(t) = \mathbf{\Gamma}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{b}(t) \tag{4}$$

where $\mathbf{z}$ is the output of the n_z sensors of the antenna array, $\mathbf{s}$ the vector of the n_s signal waveforms and $\mathbf{b}$ the additive noise. In order to explicitly show the analogy between recursive subspace identification and subspace tracking in array signal processing, it is essential to rewrite the state-space system (1) in an equivalent way to the model (4). For that, notice that Eq. (4) contains an output vector $\mathbf{z}$ composed of spatially stacked signals. Since, in identification, the available data are only temporal observations, it is therefore necessary to introduce a temporal window similar to the spatial one used in subspace tracking

$$\mathbf{y}_f(t) = \begin{bmatrix}\mathbf{y}^T(t) & \cdots & \mathbf{y}^T(t+f-1)\end{bmatrix}^T \in \mathbb{R}^{n_y f \times 1}.$$

It is also easy to verify the following relation (the stacked vectors of the input and the disturbances are defined in the same way as $\mathbf{y}_f$)

$$\mathbf{y}_f(t) = \mathbf{\Gamma}_f\mathbf{x}(t) + \mathbf{H}_f\mathbf{u}_f(t) + \mathbf{b}_f(t) \tag{5}$$

with $\mathbf{b}_f(t) = \mathbf{G}_f\mathbf{w}_f(t) + \mathbf{v}_f(t)$. The connection between subspace identification and array signal processing becomes apparent by writing

$$\mathbf{z}_f(t) = \mathbf{y}_f(t) - \mathbf{H}_f\mathbf{u}_f(t) = \mathbf{\Gamma}_f\mathbf{x}(t) + \mathbf{b}_f(t).$$

This relation stresses on both steps required to recursively estimate the extended observability matrix[1] (and by extension the state-space matrices):

1. the update of the "observation vector" $\mathbf{z}_f$ from the I/O measurements

$$\mathbf{z}_f(t) = \mathbf{y}_f(t) - \mathbf{H}_f\mathbf{u}_f(t), \tag{6}$$

2. the estimation of a basis of $\mathbf{\Gamma}_f$ from this observation vector

$$\mathbf{z}_f(t) = \mathbf{\Gamma}_f\mathbf{x}(t) + \mathbf{b}_f(t). \tag{7}$$

Both stages are now considered. The main developments concern the second step. It is important to note that it will be necessary to propose noise treatment phase so as to get consistent estimates whatever the disturbances acting on the system are. It will be more precisely shown that the introduction of an instrumental variable leads to the elimination of the noise effects.

[1] Notice that $\mathbf{H}_f$ is unknown at time t since the state-space matrices used in its construction are still not estimated.

3 Recursive estimation of the observation vector: an overview

The first step of any RSMI technique is the determination of the observation vector at each new data acquisition. By analyzing Eq. (6), it is obvious that the unknown variable is the Toeplitz matrix $\mathbf{H}_f$. This fact is the key point of the observation vector updating. Two solutions can be considered to estimate $\mathbf{z}_f$:

- to build $\mathbf{H}_f$ from the matrices estimated at the previous instant and explicitly compute the subtraction $\mathbf{y}_f(t) - \hat{\mathbf{H}}_f(t-1)\mathbf{u}_f(t)$;
- to directly estimate the observation vector from the new measured I/O data without explicitly resorting to $\mathbf{H}_f$.

The first idea was suggested by T. Gustafsson in 1997 [10]. He proposed to approximate $\mathbf{H}_f$ at time t by its estimate at time $t-1$ built from the system matrices calculated during the previous recursion. A particular matrix construction rule is formulated in [10] to reduce the computation cost. This approach is however based on an approximation which may turn out to be not fully reliable, more particularly when the system often and quickly changes.

The second solution rests on a recursive adaptation of a similar problem considered in offline subspace identification: to remove $\text{span}_{\text{col}}\{\mathbf{H}_f\}$ from the subspace $\text{span}_{\text{col}}\{\mathbf{Y}_f\}$ (see Eq. (3)). In batch subspace identification, this problem can be solved by applying an orthogonal projection of the row space of $\mathbf{Y}_f$ onto the complement of the row space of $\mathbf{U}_f$ *i.e.* $\mathbf{Y}_f\mathbf{\Pi}_{\mathbf{U}_f^\perp}$ with $\mathbf{\Pi}_{\mathbf{U}_f^\perp} = \mathbf{I}_{n_u f} - \mathbf{U}_f^T\left(\mathbf{U}_f{\mathbf{U}_f}^T\right)^{-1}\mathbf{U}_f$ [40, 41]. This projection can be computed in a stable and efficient way thanks to the following RQ factorization

$$\begin{bmatrix}\mathbf{U}_f\\ \mathbf{Y}_f\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{11} & \mathbf{0}\\ \mathbf{R}_{21} & \mathbf{R}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\end{bmatrix} \tag{8}$$

and we have $\mathbf{Y}_f\mathbf{\Pi}_{\mathbf{U}_f^\perp} = \mathbf{R}_{22}\mathbf{Q}_2$ [37]. Hence, the recursive calculation of the observation vector can be realised by developing techniques which online update $\mathbf{Y}_f\mathbf{\Pi}_{\mathbf{U}_f^\perp}$. Several algorithms have been proposed to reach this goal [29, 20]. The method considered in the following, firstly proposed by M. Verhaegen and E. Deprettere [39] and extended by M. Lovera [20], is based on the updating of the RQ factorization (8) with Givens rotations [9]. This approach has the advantage to exhibit good numerical performances with respect to the round of error in comparison with other methods [29] based on the matrix inversion lemma [9].

3.1 Updating of the ordinary MOESP RQ factorization

The basic idea of this approach is to consider the RQ decomposition (8) of the offline ordinary MOESP scheme [37] and its update at each new acquisition. For that purpose, consider the following RQ factorization

$$\begin{bmatrix}\mathbf{U}_f(\tau)\\ \mathbf{Y}_f(\tau)\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{11}(\tau) & \mathbf{0}\\ \mathbf{R}_{21}(\tau) & \mathbf{R}_{22}(\tau)\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1(\tau)\\ \mathbf{Q}_2(\tau)\end{bmatrix}.$$

When a new I/O couple $\{\mathbf{u}(\tau+1), \mathbf{y}(\tau+1)\}$ is acquired, this decomposition can be updated as[2]

$$\begin{bmatrix}\sqrt{\lambda}\begin{bmatrix}\mathbf{R}_{11}(\tau) & \mathbf{0}\\ \mathbf{R}_{21}(\tau) & \mathbf{R}_{22}(\tau)\end{bmatrix} & \begin{matrix}\mathbf{u}_f(\tau+1)\\ \mathbf{y}_f(\tau+1)\end{matrix}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1(\tau) & \mathbf{0}\\ \mathbf{Q}_2(\tau) & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}.$$

A sequence of Givens rotations $\mathbf{Rot}_G$ can then be used to annihilate the stacked input vector $\mathbf{u}_f$ and bring back the $\mathbf{R}$ factor to a block lower triangular form

$$\begin{bmatrix}\sqrt{\lambda}\begin{bmatrix}\mathbf{R}_{11}(\tau) & \mathbf{0}\\ \mathbf{R}_{21}(\tau) & \mathbf{R}_{22}(\tau)\end{bmatrix} & \begin{matrix}\mathbf{u}_f(\tau+1)\\ \mathbf{y}_f(\tau+1)\end{matrix}\end{bmatrix}\mathbf{Rot}_G(\tau+1) = \begin{bmatrix}\mathbf{R}_{11}(\tau+1) & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21}(\tau+1) & \sqrt{\lambda}\mathbf{R}_{22}(\tau) & \breve{\mathbf{z}}_f(\tau+1)\end{bmatrix}.$$

$\breve{\mathbf{z}}_f$ is the vector obtained by modifying $\mathbf{y}_f$ in order to include the information contained in $\mathbf{u}_f$ and $\begin{bmatrix}\mathbf{R}_{11}^T & \mathbf{R}_{21}^T\end{bmatrix}^T$. It is possible to prove that $\breve{\mathbf{z}}_f(t) = \pm\mathbf{z}_f(t)$. [21] .

[2] The forgetting factor λ is introduced to weight the past informations.

Remark 1 *The theoretical growth size of the* $\mathbf{R}$ *factor is not a problem in practice. Indeed,* $\mathbf{R}_{11}$ *and* $\mathbf{R}_{21}$ *are only used to estimate* $\mathbf{z}_f$*. Thus, there is no need to update* $\mathbf{R}_{22}$*. The computational complexity of this partial RQ factorization update is* $\mathcal{O}\left(\left(\frac{n_u}{2}+n_y\right)n_u f^2\right)$*.*

This technique leads to unbiased estimates only if the output measurement noise $\mathbf{v}$ is white and the process noise $\mathbf{w}$ is null. Since this condition can be too restrictive in practice, the development of a recursive version of the PI/PO MOESP RQ factorization has been suggested in [20] to estimate an asymptotically noise purged observation vector in a coloured noisy framework.

3.2 Updating of the PI/PO MOESP RQ factorization

The basic idea of the method is based on the same steps as in the previous paragraph *i.e.* updating a particular RQ factorization at each time instant by applying Givens rotations. More precisely, in the batch subspace identification framework, when $\mathbf{w} \equiv \mathbf{0}$ and $\mathbf{v}$ is an ergodic sequence independent of the input signal or when $(\mathbf{v}, \mathbf{w})$ are ergodic sequences satisfying (2), the cancellation of $\text{span}_{\text{col}}\{\mathbf{H}_f\}$ and $\text{span}_{\text{col}}\{\mathbf{B}_f\}$ from $\text{span}_{\text{col}}\{\mathbf{Y}_f\}$ is achieved by introducing the RQ decomposition [38]

$$\begin{bmatrix}\mathbf{U}_f\\ \mathbf{\Xi}\\ \mathbf{Y}_f\end{bmatrix} = \begin{bmatrix}\mathbf{R}_{11} & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{0}\\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\\ \mathbf{Q}_3\end{bmatrix}$$

where $\mathbf{\Xi}$ is an instrumental variable[3] such that $\lim_{M\to\infty}\frac{1}{M}\mathbf{B}_f\mathbf{\Xi}^T = \mathbf{0}$ and $\text{rank}\left\{\mathbf{X}\mathbf{\Xi}^T\right\} = n_x$. Indeed, it is nowadays well known that, under these assumptions [38]

$$\lim_{M\to\infty}\frac{1}{\sqrt{M}}\mathbf{R}_{32}\mathbf{Q}_2 = \lim_{M\to\infty}\frac{1}{\sqrt{M}}\mathbf{\Gamma}_f\mathbf{X}. \tag{9}$$

When a new I/O couple is acquired, this decomposition is completed as follows

$$\begin{bmatrix}\sqrt{\lambda}\begin{bmatrix}\mathbf{R}_{11}(\tau) & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21}(\tau) & \mathbf{R}_{22}(\tau) & \mathbf{0}\\ \mathbf{R}_{31}(\tau) & \mathbf{R}_{32}(\tau) & \mathbf{R}_{33}(\tau)\end{bmatrix} & \begin{matrix}\mathbf{u}_f(\tau+1)\\ \boldsymbol{\xi}(\tau+1)\\ \mathbf{y}_f(\tau+1)\end{matrix}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_1(\tau) & \mathbf{0}\\ \mathbf{Q}_2(\tau) & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}.$$

As in Subsection 3.1, Givens rotations can be used to update this factorization. More precisely, two sequences of Givens rotations are applied to bring back the $\mathbf{R}$ factor to the following block lower triangular form

$$\begin{aligned}&\begin{bmatrix}\sqrt{\lambda}\begin{bmatrix}\mathbf{R}_{11}(\tau) & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21}(\tau) & \mathbf{R}_{22}(\tau) & \mathbf{0}\\ \mathbf{R}_{31}(\tau) & \mathbf{R}_{32}(\tau) & \mathbf{R}_{33}(\tau)\end{bmatrix} & \begin{matrix}\mathbf{u}_f(\tau+1)\\ \boldsymbol{\xi}(\tau+1)\\ \mathbf{y}_f(\tau+1)\end{matrix}\end{bmatrix}\mathbf{Rot}_{G_1}(\tau+1)\mathbf{Rot}_{G_2}(\tau+1) =\\ &\qquad\begin{bmatrix}\mathbf{R}_{11}(\tau+1) & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21}(\tau+1) & \sqrt{\lambda}\begin{bmatrix}\mathbf{R}_{22}(\tau) & \mathbf{0}\\ \mathbf{R}_{32}(\tau) & \mathbf{R}_{33}(\tau)\end{bmatrix} & & \begin{matrix}\boldsymbol{\xi}(\tau+1)\\ \check{\mathbf{z}}_f(\tau+1)\end{matrix}\\ \mathbf{R}_{31}(\tau+1) & & & \end{bmatrix}\mathbf{Rot}_{G_2}(\tau+1) =\\ &\qquad\qquad\begin{bmatrix}\mathbf{R}_{11}(\tau+1) & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{21}(\tau+1) & \mathbf{R}_{22}(\tau+1) & \mathbf{0} & \mathbf{0}\\ \mathbf{R}_{31}(\tau+1) & \mathbf{R}_{32}(\tau+1) & \sqrt{\lambda}\mathbf{R}_{33}(\tau) & \check{\check{\mathbf{z}}}_f(\tau+1)\end{bmatrix}.\end{aligned} \tag{10}$$

In order to stress on the importance of this update for RSMI, it is necessary to show the link between the signal subspace and the vectors $\check{\mathbf{z}}_f$ and $\check{\check{\mathbf{z}}}_f$. This can be realised by noticing first of all that [18]

$$\mathbf{R}_{32}(t)\mathbf{R}_{32}^T(t) = \lambda\mathbf{R}_{32}(t-1)\mathbf{R}_{32}^T(t-1) + \check{\mathbf{z}}_f(t)\check{\mathbf{z}}_f^T(t) - \check{\check{\mathbf{z}}}_f(t)\check{\check{\mathbf{z}}}_f^T(t).$$

Now, according to Eq. (9), we have

$$\lim_{M\to\infty}\frac{1}{M}\mathbf{R}_{32}\mathbf{R}_{32}^T = \lim_{M\to\infty}\frac{1}{M}\mathbf{\Gamma}_f\mathbf{X}\mathbf{X}^T\mathbf{\Gamma}_f^T.$$

[3] The instruments are chosen as past inputs in the PI scheme while past inputs and past outputs are used in the PO scheme [38].

Then, it follows that

$$\mathbb{E}\left\{\check{\mathbf{z}}_f(t)\check{\mathbf{z}}_f^T(t) - \check{\check{\mathbf{z}}}_f(t)\check{\check{\mathbf{z}}}_f^T(t)\right\} = \mathbf{\Gamma}_f \mathbf{R}_{\mathbf{x}} \mathbf{\Gamma}_f^T.$$

This last Eq. points out that Eq. (10) asymptotically leads to a covariance matrix from which the subspace spanned by the columns of the extended observability matrix can be extracted. In the following, this covariance matrix will be denoted as $\mathbf{R}_{\mathbf{z}_f}$. In practice, the sample covariance matrix $\hat{\mathbf{R}}_{\mathbf{z}_f}$ is updated via the recursive formulation

$$\hat{\mathbf{R}}_{\mathbf{z}_f}(t) = \lambda \hat{\mathbf{R}}_{\mathbf{z}_f}(t-1) + \check{\mathbf{z}}_f(t)\check{\mathbf{z}}_f^T(t) - \check{\check{\mathbf{z}}}_f(t)\check{\check{\mathbf{z}}}_f^T(t). \tag{11}$$

Remark 2 *As previously, there is no need to totally complete the PI/PO MOESP RQ factorization in practice. The complexity of this update technique is in* $\mathcal{O}\left(\left(\frac{n_u f + n_\xi}{2} + n_y f\right)(n_u f + n_\xi)\right)$.

4 Recursive estimation of an observability matrix basis

The previous section has introduced two recursive algorithms to update the observation vector. The second step of the recursive subspace identification methods developed in this paper consists in online estimating a basis of the observability subspace. The estimation algorithms proposed in the following are based on the adaptation of a particular array signal processing technique: the propagator method [27]. This technique has the advantage of supplying a decomposition of the observation space into two complementary subspaces via the use of a simple linear operator. Furthermore, its adjustment to the recursive identification problem leads to quadratic criteria needing neither approximation nor constraint. Several recursive algorithms can be developed according to the hypotheses fixed on the signals acting on the system. Seven of them will be introduced in this section. It will be more precisely emphasised that most of them can consistently estimate a basis of the observability subspace of systems subjected to coloured unknown disturbances.

4.1 Adaptation of the propagator method to identification

4.1.1 Problem formulation

Assume that $\{\mathbf{A}, \mathbf{C}\}$ is observable. Then, since $\mathbf{\Gamma}_f \in \mathbb{R}^{n_y f \times n_x}$ with $n_y f > n_x$, the observability matrix has, at least, n_x linearly independent rows. Under the hypothesis that the order n_x is *a priori* known, it is possible to build a permutation matrix $\mathbf{S} \in \mathbb{R}^{n_y f \times n_y f}$ such that the extended observability matrix can be decomposed in the following way

$$\mathbf{S}\mathbf{\Gamma}_f = \begin{bmatrix} \mathbf{\Gamma}_{f_1} \\ \mathbf{\Gamma}_{f_2} \end{bmatrix} \begin{matrix} \}\mathbb{R}^{n_x \times n_x} \\ \}\mathbb{R}^{n_y f - n_x \times n_x} \end{matrix}$$

where $\mathbf{\Gamma}_{f_1}$ is the block of n_x independent rows and $\mathbf{\Gamma}_{f_2}$ the matrix of the $n_y f - n_x$ others. By construction, $\mathbf{\Gamma}_{f_2}$ can be expressed as a linear combination of $\mathbf{\Gamma}_{f_1}$. More particularly, there is a unique operator $\mathbf{P}_f \in \mathbb{R}^{n_x \times n_y f - n_x}$ named propagator [27] such that $\mathbf{\Gamma}_{f_2} = \mathbf{P}_f^T \mathbf{\Gamma}_{f_1}$. It is also easy to verify that

$$\mathbf{S}\mathbf{\Gamma}_f = \begin{bmatrix} \mathbf{\Gamma}_{f_1} \\ \mathbf{\Gamma}_{f_2} \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}_{f_1} \\ \mathbf{P}_f^T \mathbf{\Gamma}_{f_1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{P}_f^T \end{bmatrix}}_{\mathbf{E}_o} \mathbf{\Gamma}_{f_1}.$$

This proves that the columns of $\mathbf{S}\mathbf{\Gamma}_f$ are linear combinations of $\mathbf{E}_o$. Now, since $\operatorname{rank}(\mathbf{\Gamma}_{f_1}) = n_x$, $\operatorname{span}_{\text{col}}(\mathbf{S}\mathbf{\Gamma}_f) = \operatorname{span}_{\text{col}}(\mathbf{E}_o)$. This relation implies that it is possible to determine the observability matrix in a particular basis from $\mathbf{E}_o$. Hence, the computation of such a matrix only requires to estimate the propagator. Thus, assuming that the order n_x is *a priori* known and the system is observable, an estimate of the subspace spanned by the observability matrix is available by estimating $\mathbf{P}_f$. This problem is considered in the following paragraphs. However, before developing algorithms for the recursive propagator computation, the problem of the data reorganization in order to make sure that the first n_x rows of the observability matrix are linearly independent has to be investigated.

4.1.2 Discussion on the matrix S construction

The use of the propagator for the observability matrix estimation is based on the following challenge: finding the permutation matrix $\mathbf{S}$, without knowing $\mathbf{\Gamma}_f$, such that the first n_x rows of $\mathbf{S}\mathbf{\Gamma}_f$ are linearly independent. As long as observable MISO systems are considered, $\mathbf{S}$ can be trivially chosen as the identity matrix since, in this case, the first n_x rows of $\mathbf{\Gamma}_f$ are linearly independent by construction. For the general MIMO class of systems, the selection of this matrix $\mathbf{S}$ becomes slightly complicated. In order to well understand the problem, let denote the rows of the extended observability matrix related to the output j as follows

$$\boldsymbol{\gamma}_j = \begin{bmatrix} \mathbf{c}_j & \mathbf{A}^T\mathbf{c}_j & \cdots & \left(\mathbf{A}^{f-1}\right)^T\mathbf{c}_j \end{bmatrix}^T$$

where $\mathbf{c}_j^T$ refers to the jth row of $\mathbf{C}$. If all the poles are observable from the output y_j, (as it is for a MISO system), then $\boldsymbol{\gamma}_j$ is necessarily of rank n_x. But this property is far from being true for any output j of a general MIMO system. In fact, some dynamics in the state vector could not be available through y_j. To find a solution to this problem, consider an auxiliary output $\bar{y}$ defined as a combination of the system outputs $\bar{y} = \sum_{j=1}^{n_y} \alpha_j\, y_j$ with α_j real numbers. The goal of this manipulation is to replace in the identification procedure one of the outputs (for example y_1) by this auxiliary output from which all the poles of the system are observable. The selection of the coefficients α_j must guarantee the observability of all the system poles in $\bar{y}$, in other words $\bar{\boldsymbol{\gamma}} = \sum_{j=1}^{n_y} \alpha_j \boldsymbol{\gamma}_j$ has to be of full rank n_x. The coefficients α_j can be generated randomly as nonzero real numbers. It could be noticed that *a priori* knowledge about the repartition of the system dynamics can also be used for this choice.

After introducing this auxiliary output, the data equation (5) can be rewritten with some modifications as follows

$$\bar{\mathbf{y}}_f(t) = \bar{\mathbf{\Gamma}}_f\mathbf{x}(t) + \bar{\mathbf{H}}_f\mathbf{u}_f(t) + \bar{\mathbf{b}}_f(t)$$

with $\bar{\mathbf{y}}_f(t)$ is the new output vector in which $\bar{y}(t)$ has been substituted for the first component and with

$$\bar{\mathbf{\Gamma}}_f = \begin{bmatrix} \bar{\mathbf{C}} \\ \bar{\mathbf{C}}\mathbf{A} \\ \vdots \\ \bar{\mathbf{C}}\mathbf{A}^{f-1} \end{bmatrix} \text{ and } \bar{\mathbf{H}}_f = \begin{bmatrix} \bar{\mathbf{D}} & \mathbf{0} & \cdots & \mathbf{0} \\ \bar{\mathbf{C}}\mathbf{B} & \bar{\mathbf{D}} & \cdots & \mathbf{0} \\ \bar{\mathbf{C}}\mathbf{A}\mathbf{B} & \bar{\mathbf{C}}\mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{C}}\mathbf{A}^{f-2}\mathbf{B} & \bar{\mathbf{C}}\mathbf{A}^{f-3}\mathbf{B} & \cdots & \bar{\mathbf{D}} \end{bmatrix}$$

where $\bar{\mathbf{C}}$ refers to the matrix $\mathbf{C}$ with its first row replaced by $\bar{\mathbf{c}}_1^T = \sum_{j=1}^{n_y} \alpha_j \mathbf{c}_j^T$ and where $\bar{\mathbf{D}}$ is defined in a similar way. Then, it becomes straightforward to built the permutation matrix as follows

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_{n_y f}(1,:) \\ \mathbf{I}_{n_y f}(n_y+1,:) \\ \vdots \\ \mathbf{I}_{n_y f}\left((f-1)n_y+1,:\right) \\ \mathbf{I}_{n_y f}(2:n_y,:) \\ \mathbf{I}_{n_y f}(n_y+2:2n_y,:) \\ \vdots \\ \mathbf{I}_{n_y f}\left((f-1)n_y+2:fn_y,:\right) \end{bmatrix}.$$

Remark 3 *In the remainder of the paper we will still use the notations of Eq.* (5) *even after the transformation suggested above. Knowing the permutation matrix* $\mathbf{S}$*, the matrix* $\hat{\mathbf{\Gamma}}_f$ *can be deduced from* $\hat{\mathbf{P}}_f$ *by pre-multiplying* $\hat{\mathbf{E}}_o$ *by* $\mathbf{S}^T$ *since* $\mathbf{S}^T\mathbf{S} = \mathbf{I}$.

The problem of the propagator estimation is considered in the following of this section. The effects of the noise on the estimates reliability are more precisely analysed.

4.1.3 Minimised criterion in the noise free case

Consider Eq. (7) and assume that the noise is identically null. After the application of the transformation matrix $\mathbf{S}$ such that the first n_x rows of $\mathbf{\Gamma}_f$ are linearly independent, the following partition of the observation vector can be introduced

$$\mathbf{z}_f(t) = \begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{P}_f^T \end{bmatrix} \mathbf{\Gamma}_{f_1} \mathbf{x}(t) = \begin{bmatrix} \mathbf{z}_{f_1}(t) \\ \mathbf{z}_{f_2}(t) \end{bmatrix} \begin{matrix} \} \in \mathbb{R}^{n_x \times 1} \\ \} \in \mathbb{R}^{n_y f - n_x \times 1} \end{matrix} \tag{12}$$

where $\mathbf{z}_{f_1}$ and $\mathbf{z}_{f_2}$ are the components of $\mathbf{z}_f$ corresponding respectively to the n_x rows of $\mathbf{\Gamma}_{f_1}$ and $n_y f - n_x$ rows of $\mathbf{\Gamma}_{f_2}$ (the same symbols are used before and after the reorganization for the sake of simplicity). In the ideal noise free case, it is easy to show that $\mathbf{z}_{f_2} = \mathbf{P}_f^T \mathbf{z}_{f_1}$ since $\mathbf{\Gamma}_{f_1}$ is invertible. Then, the propagator can be estimated by minimising the following quadratic criterion

$$J(\mathbf{P}_f) = \mathbb{E}\left\| \mathbf{z}_{f_2} - \mathbf{P}_f^T \mathbf{z}_{f_1} \right\|^2, \tag{13}$$

the (asymptotic) least squares optimum of which is given by [17]

$$\hat{\mathbf{P}}_f^T = \mathbf{R}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}} \mathbf{R}_{\mathbf{z}_{f_1}}^{-1}. \tag{14}$$

Remark 4 *Notice that the uniqueness of $\hat{\mathbf{P}}_f$ is ensured by the convexity of the criterion* (13)*, which, in turn, can be guaranteed by suitable persistency of excitation assumptions (see [25] for the convergence conditions of the propagator-based methods in the recursive identification framework).*

In order to develop a recursive minimization algorithm and since only a finite number of data is available in practice, write the cost function (13) as follows

$$J(\mathbf{P}_f(t)) = \sum_{k=1}^{t} \beta^{t-k} \left\| \mathbf{z}_{f_2}(k) - \mathbf{P}_f^T(t) \mathbf{z}_{f_1}(k) \right\|^2 \tag{15}$$

obtained by replacing the expectation operator with a finite exponentially weighted sum. The minimization of such a criterion is recursively feasible by applying a classic recursive least squares approach [17]. The outcome algorithm, named RPM1, and the computational load of each step are summarised in Table 1.

```
Build $\check{\mathbf{L}}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \ldots$
    $\check{\mathbf{K}}_f(t) = \dfrac{\mathbf{z}_{f_1}^T(t)\check{\mathbf{L}}_f(t-1)}{\beta + \mathbf{z}_{f_1}^T(t)\check{\mathbf{L}}_f(t-1)\mathbf{z}_{f_1}(t)}$        $n_x(n_x+1)$
    $\check{\mathbf{L}}_f(t) := \mathbf{R}_{\mathbf{z}_{f_1}}^{-1}(t)$
                 $= \dfrac{1}{\beta}\left(\check{\mathbf{L}}_f(t-1) - \check{\mathbf{L}}_f(t-1)\mathbf{z}_{f_1}(t)\check{\mathbf{K}}_f(t)\right)$        $n_x^2$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{z}_{f_2}(t) - \mathbf{P}_f^T(t-1)\mathbf{z}_{f_1}(t)\right)\check{\mathbf{K}}_f(t)$        $2n_x(n_y f - n_x)$
end
```

Table 1: The RPM1 algorithm and its computational load.

4.1.4 Noise effects

Assume now that the noise $\mathbf{b}_f$ is no more null. Then, it is easy to show that the LS solution (14) leads to a biased estimate, even if the residual vector $\mathbf{b}_f$ is zero mean spatially and temporally white. In order to emphasise this drawback, apply first of all to $\mathbf{b}_f$ a reorganization similar to (12)

$$\mathbf{b}_f = \begin{bmatrix} \mathbf{b}_{f_1} \\ \mathbf{b}_{f_2} \end{bmatrix} \begin{matrix} \} \in \mathbb{R}^{n_x \times 1} \\ \} \in \mathbb{R}^{n_y f - n_x \times 1} \end{matrix} .$$

Then, the noise covariance matrix is

$$\mathbf{R}_{\mathbf{b}_f} = \begin{bmatrix} \mathbf{R}_{\mathbf{b}_{f_1}} & \mathbf{R}_{\mathbf{b}_{f_1}\mathbf{b}_{f_2}} \\ \mathbf{R}_{\mathbf{b}_{f_2}\mathbf{b}_{f_1}} & \mathbf{R}_{\mathbf{b}_{f_2}} \end{bmatrix}.$$

Hence, it is possible to write the covariance matrix of $\mathbf{z}_f$ as

$$\mathbf{R}_{\mathbf{z}_f} = \begin{bmatrix} \mathbf{R}_{\mathbf{z}_{f_1}} & \mathbf{R}_{\mathbf{z}_{f_1}\mathbf{z}_{f_2}} \\ \mathbf{R}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}} & \mathbf{R}_{\mathbf{z}_{f_2}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{P}_f^T \end{bmatrix} \underbrace{\mathbf{\Gamma}_{f_1}\mathbf{R}_{\mathbf{x}}\mathbf{\Gamma}_{f_1}^T}_{\mathbf{R}_{\bar{\mathbf{x}}}} \begin{bmatrix} \mathbf{I}_{n_x} & \mathbf{P}_f \end{bmatrix} + \begin{bmatrix} \mathbf{R}_{\mathbf{b}_{f_1}} & \mathbf{R}_{\mathbf{b}_{f_1}\mathbf{b}_{f_2}} \\ \mathbf{R}_{\mathbf{b}_{f_2}\mathbf{b}_{f_1}} & \mathbf{R}_{\mathbf{b}_{f_2}} \end{bmatrix}$$

with[4] $\bar{\mathbf{x}} = \mathbf{\Gamma}_{f_1}\mathbf{x}$. Thus, the following relations are verified

$$\begin{aligned} \mathbf{R}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}} &= \mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}} + \mathbf{R}_{\mathbf{b}_{f_2}\mathbf{b}_{f_1}} = \mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}} \\ \mathbf{R}_{\mathbf{z}_{f_1}} &= \mathbf{R}_{\bar{\mathbf{x}}} + \mathbf{R}_{\mathbf{b}_{f_1}} = \mathbf{R}_{\bar{\mathbf{x}}} + \sigma^2\mathbf{I}_{n_x} \end{aligned}$$

when $\mathbf{R}_{\mathbf{b}_f} = \sigma^2\mathbf{I}_{n_y f}$. Then, we get

$$\hat{\mathbf{P}}_f^T = \mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}}\left(\mathbf{R}_{\bar{\mathbf{x}}} + \sigma^2\mathbf{I}_{n_x}\right)^{-1} \neq \mathbf{P}_f^T, \tag{16}$$

equation which shows that the propagator is asymptotically biased. This difficulty could be partially circumvented by considering a Total Least Squares [35] approach to the problem. However this would likely complicate the recursive implementation of the solution. Two other solutions can be proposed

- The first one assumes that the observation vector has been estimated via the recursive version of the PI/PO MOESP schemes (see §3.2). Then, an asymptotically noise free covariance matrix $\mathbf{R}_{\mathbf{z}_f}$ is computed with the property to be linked to the observability matrix by the relation

$$\mathbf{R}_{\mathbf{z}_f} = \mathbf{\Gamma}_f\mathbf{R}_{\mathbf{x}}\mathbf{\Gamma}_f^T. \tag{17}$$

- The second approach assumes that the observation vector has been computed by using the recursive version of ordinary MOESP (see §3.1). Then, no disturbance treatment has ever been realized. The solution consists in introducing an instrumental variable in the criterion (13) so that the new cost function is usable whatever the colour of the noise acting on the system is.

Both approaches will be respectively considered in Subsection 4.2 and 4.3. The algorithms developed in these paragraphs have more particularly the property to supply consistent estimates of the extended observability matrix in a noisy framework.

Remark 5 *It is interesting to complete this study by quantifying the outcomes of the disturbances in the propagator estimation. The perturbation theory [34] allows us to write that, for all matrix* $\mathbf{M}$ *and small perturbation* $\Delta\mathbf{M}$

$$(\mathbf{M} + \Delta\mathbf{M})^{-1} \cong \mathbf{M}^{-1} - \mathbf{M}^{-1}\Delta\mathbf{M}\mathbf{M}^{-1}.$$

By writing $\mathbf{M} = \mathbf{R}_{\bar{\mathbf{x}}}$ *and* $\mathbf{\Delta M} = \sigma^2\mathbf{I}_{n_x}$*, it is straightforward to show that*

$$\left(\mathbf{R}_{\bar{\mathbf{x}}} + \sigma^2\mathbf{I}_{n_x}\right)^{-1} = \mathbf{R}_{\bar{\mathbf{x}}}^{-1} - \sigma^2\mathbf{R}_{\bar{\mathbf{x}}}^{-1}\mathbf{R}_{\bar{\mathbf{x}}}^{-1}.$$

Equation (16) *can then be written as*

$$\hat{\mathbf{P}}_f^T \cong \mathbf{P}_f^T - \sigma^2\mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}}^{-1}.$$

This clearly proves that the propagator is asymptotically biased.

4.2 Estimation from the recursive PI/PO MOESP scheme update

Assume that an asymptotically noise free covariance matrix $\mathbf{R}_{\mathbf{z}_f}$ has been estimated via the recursive version of the PI/PO MOESP scheme (see §3.2). Then, the estimation of the propagator is no more feasible from the criterion (13) since this one requires $\mathbf{z}_f$ which is not available with the Eq. (11). So, it is necessary to propose other cost functions which take into account the matrix $\mathbf{R}_{\mathbf{z}_f}$ instead of the observation vector.

[4] Notice that $\mathbf{\Gamma}_{f_1}$ can be considered as a similarity transformation since rank $\{\mathbf{\Gamma}_{f_1}\} = n_x$.

4.2.1 RPM2 algorithm

In order to determine a new criterion, consider the expression of the observability matrix in the propagator basis

$$\mathbf{\Gamma}_f = \begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{P}_f^T \end{bmatrix} \mathbf{\Gamma}_{f_1}.$$

Then, Eq. (17) can be rewritten as follows

$$\mathbf{R}_{\mathbf{z}_f} = \begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{P}_f^T \end{bmatrix} \mathbf{R}_{\bar{\mathbf{x}}} \begin{bmatrix} \mathbf{I}_{n_x} & \mathbf{P}_f \end{bmatrix}.$$

Thus, by construction,

$$\begin{aligned} \mathbf{R}_{\mathbf{z}_f} &= \begin{bmatrix} \mathbf{R}_{\check{\mathbf{z}}_{f_1}}(t) - \mathbf{R}_{\check{\check{\mathbf{z}}}_{f_1}}(t) & \mathbf{R}_{\check{\mathbf{z}}_{f_1}\check{\mathbf{z}}_{f_2}}(t) - \mathbf{R}_{\check{\check{\mathbf{z}}}_{f_1}\check{\check{\mathbf{z}}}_{f_2}}(t) \\ \mathbf{R}_{\check{\mathbf{z}}_{f_2}\check{\mathbf{z}}_{f_1}}(t) - \mathbf{R}_{\check{\check{\mathbf{z}}}_{f_2}\check{\check{\mathbf{z}}}_{f_1}}(t) & \mathbf{R}_{\check{\mathbf{z}}_{f_2}} - \mathbf{R}_{\check{\check{\mathbf{z}}}_{f_2}}(t) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{\bar{\mathbf{x}}} & \mathbf{R}_{\bar{\mathbf{x}}}\mathbf{P}_f \\ \mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}} & \mathbf{P}_f^T\mathbf{R}_{\bar{\mathbf{x}}}\mathbf{P}_f \end{bmatrix}. \end{aligned} \tag{18}$$

This relation shows that the propagator can be estimated by minimising the following cost function

$$\bar{J}(\mathbf{P}_f(t)) = \left\| \left(\hat{\mathbf{R}}_{\check{\mathbf{z}}_{f_2}\check{\mathbf{z}}_{f_1}}(t) - \hat{\mathbf{R}}_{\check{\check{\mathbf{z}}}_{f_2}\check{\check{\mathbf{z}}}_{f_1}}(t)\right) - \mathbf{P}_f^T(t)\left(\hat{\mathbf{R}}_{\check{\mathbf{z}}_{f_1}}(t) - \hat{\mathbf{R}}_{\check{\check{\mathbf{z}}}_{f_1}}(t)\right) \right\|_F^2 \tag{19}$$

since, by assuming that the involved inverse matrices exist (see [25] for conditions of invertibility), its minimising argument is given by

$$\hat{\mathbf{P}}_f^T(t) = \left(\hat{\mathbf{R}}_{\check{\mathbf{z}}_{f_2}\check{\mathbf{z}}_{f_1}}(t) - \hat{\mathbf{R}}_{\check{\check{\mathbf{z}}}_{f_2}\check{\check{\mathbf{z}}}_{f_1}}(t)\right)\left(\hat{\mathbf{R}}_{\check{\mathbf{z}}_{f_1}}(t) - \hat{\mathbf{R}}_{\check{\check{\mathbf{z}}}_{f_1}}(t)\right)^{-1}. \tag{20}$$

An RLS-based algorithm can be developed to recursively estimate this optimum. The algorithm introduced in Table 2 and named RPM2 summarises the computational steps of the recursive formulation of (20).

```
Build $\underline{\mathbf{L}}_f(0)$, $\underline{\underline{\mathbf{L}}}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \dots$
    $\underline{\mathbf{L}}_f(t) = \frac{1}{\lambda}\left(\underline{\underline{\mathbf{L}}}_f(t-1) - \frac{\underline{\underline{\mathbf{L}}}_f(t-1)\check{\mathbf{z}}_{f_1}(t)\check{\mathbf{z}}_{f_1}^T(t)\underline{\underline{\mathbf{L}}}_f(t-1)}{\lambda + \check{\mathbf{z}}_{f_1}^T(t)\underline{\underline{\mathbf{L}}}_f(t-1)\check{\mathbf{z}}_{f_1}(t)}\right)$        $n_x(2n_x+1)$
    $\underline{\underline{\mathbf{L}}}_f(t) = \underline{\mathbf{L}}_f(t) + \frac{\underline{\mathbf{L}}_f(t)\check{\check{\mathbf{z}}}_{f_1}(t)\check{\check{\mathbf{z}}}_{f_1}{}^T(t)\underline{\mathbf{L}}_f(t)}{1 - \check{\check{\mathbf{z}}}_{f_1}{}^T(t)\underline{\mathbf{L}}_f(t)\check{\check{\mathbf{z}}}_{f_1}(t)}$        $n_x(2n_x+1)$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\check{\mathbf{z}}_{f_2}(t) - \mathbf{P}_f^T(t-1)\check{\mathbf{z}}_{f_1}(t)\right)\check{\mathbf{z}}_{f_1}^T(t)\underline{\mathbf{L}}_f(t)$
        $- \left(\check{\check{\mathbf{z}}}_{f_2}(t) - \mathbf{P}_f^T(t-1)\check{\check{\mathbf{z}}}_{f_1}(t)\right)\check{\check{\mathbf{z}}}_{f_1}^T(t)\underline{\underline{\mathbf{L}}}_f(t)$        $2n_x(2n_y f - n_x)$
end
```

Table 2: The RPM2 algorithm and its computational load.

4.2.2 COPM algorithm

In order to avoid the use of the matrix inversion lemma[5] twice, a second approach can be introduced. More precisely, by writing the decomposition of $\mathbf{R}_{\mathbf{z}_f}$ (see Eq. (18)) in a compact manner as follows

$$\mathbf{R}_{\mathbf{z}_f} = \begin{bmatrix} \mathbf{R}_{\mathbf{z}_{f_1}} & \mathbf{R}_{\mathbf{z}_{f_1}\mathbf{z}_{f_2}} \\ \mathbf{R}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}} & \mathbf{R}_{\mathbf{z}_{f_2}} \end{bmatrix}, \tag{21}$$

[5] This technique is known to suffer from numerical problems when ill-conditioned matrices are considered.

the cost function (19) becomes

$$\bar{J}(\mathbf{P}_f(t)) = \left\| \hat{\mathbf{R}}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(t) - \mathbf{P}_f^T(t)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}}(t) \right\|_F^2 . \tag{22}$$

Then, by definition of the Frobenius matrix norm [9], the criterion (22) can be modified as

$$\bar{J}(\mathbf{P}_f(t)) = \sum_{i=1}^{n_x} \left\| \mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(t) - \mathbf{P}_f^T(t)\mathbf{r}^i_{\mathbf{z}_{f_1}}(t) \right\|^2$$

where $\mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}$ and $\mathbf{r}^i_{\mathbf{z}_{f_1}}$ are respectively the ith columns of $\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}$ and $\hat{\mathbf{R}}_{\mathbf{z}_{f_1}}$. A sufficient condition to the minimization of the global cost function $\bar{J}$ is the sequential optimization of

$$\bar{J}_i(\mathbf{P}_f(t)) = \left\| \mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(t) - \mathbf{P}_f^T(t)\mathbf{r}^i_{\mathbf{z}_{f_1}}(t) \right\|^2 \text{ for } i \in \{1, \cdots, n_x\} . \tag{23}$$

By being inspired by the works of J. L. Yu [45] on the COPAST method, in order to improve the numerical stability of the minimization algorithm applied to (23) and to introduce forgetting capacities for tracking, the cost function $\bar{J}_i$ can be modified as follows

$$\bar{J}_i(\mathbf{P}_f(t)) = \sum_{k=1}^{t} \beta^{t-k} \left\| \mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(k) - \mathbf{P}_f^T(t)\mathbf{r}^i_{\mathbf{z}_{f_1}}(k) \right\|^2 \text{ for } i \in \{1, \cdots, n_x\} .$$

Each cost function $\bar{J}_i$ is then equivalent to the criterion (15). The RPM1 method can therefore be adjusted to this optimization problem. The corresponding algorithm, named COPM, is introduced in Table 3.

```
Build $\bar{\mathbf{L}}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \ldots$
    $\hat{\mathbf{R}}_{\mathbf{z}_f}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_f}(t-1) + \check{\mathbf{z}}_f(t)\check{\mathbf{z}}_f^T(t) - \check{\check{\mathbf{z}}}_f(t)\check{\check{\mathbf{z}}}_f^T(t)$        $2(n_y f)^2$
    $i = \text{remainder}(t, n_x) + 1$
    $\mathbf{r}^i_{\mathbf{z}_{f_1}}(t) = \hat{\mathbf{R}}_{\mathbf{z}_f}(1 : n_x, i)(t)$
    $\mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(t) = \hat{\mathbf{R}}_{\mathbf{z}_f}(n_x + 1 : n_y f, i)(t)$
    $\bar{\mathbf{K}}_f(t) = \dfrac{\mathbf{r}^{i\,T}_{\mathbf{z}_{f_1}}(t)\bar{\mathbf{L}}_f(t-1)}{\beta + \mathbf{r}^{i\,T}_{\mathbf{z}_{f_1}}(t)\bar{\mathbf{L}}_f(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}}(t)}$        $n_x(n_x + 1)$
    $\bar{\mathbf{L}}_f(t) = \dfrac{1}{\beta}\left(\bar{\mathbf{L}}_f(t-1) - \bar{\mathbf{L}}_f(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}}(t)\bar{\mathbf{K}}_f(t)\right)$        $n_x^2$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{r}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}(t) - \mathbf{P}_f^T(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}}(t)\right)\bar{\mathbf{K}}_f(t)$        $2n_x(n_y f - n_x)$
end
```

Table 3: The COPM algorithm and its computational load.

Note that this algorithm requires to extract the vectors $\hat{\mathbf{r}}^i_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}$ and $\hat{\mathbf{r}}^i_{\mathbf{z}_{f_1}}$. As shown in Table 3, these ones are obtained at each update of the covariance matrix $\hat{\mathbf{R}}_{\mathbf{z}_f}$ by noticing that the matrices $\hat{\mathbf{R}}_{\mathbf{z}_{f_1}}$ and $\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\mathbf{z}_{f_1}}$ respectively correspond to the n_x first rows and $n_y f - n_x$ last rows of the n_x first columns of $\hat{\mathbf{R}}_{\mathbf{z}_f}$ (see Eq. (18) and (21)).

4.3 Estimation from the ordinary MOESP scheme update

The online propagator updating techniques developed in Subsection 4.2 exclusively employ the covariance matrix $\mathbf{R}_{\mathbf{z}_f}$. As specified in the Subsections 3.1 and 3.2, the computational cost of the covariance matrix update is greater than the recursive estimation of the observation vector. Thus, it would be interesting to propose recursive propagator estimation techniques which use the solution of the ordinary MOESP RQ factorization updating. The study of the noise effects proposed in the paragraph 4.1.4 has shown that the the propagator approach provides biased estimates even in a white noise framework. Since the algorithm proposed in §3.1 does not treat the disturbances effects, it is now essential to introduce recursive methods which circumvent this problem.

Many identification methods using the instrumental variable approach [33, 17] have been developed since the 70's to model systems in noisy environment. Their principle consists in introducing into the minimised criterion a vector, composed for example by I/O delayed data, in order to zero out the influence of the noise contained in the regressors at time t. In order to employ such a technique for our identification problem, consider an instrumental variable $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi \times 1}$, $n_\xi \geq n_x$, assumed to be uncorrelated with the noise $\mathbf{b}_f$ but sufficiently correlated with the state vector $\mathbf{x}$. By introducing this vector into the cost function (13) we get

$$J_{IV}(\mathbf{P}_f(t)) = \left\| \hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) - \mathbf{P}_f^T(t)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t) \right\|_F^2 \tag{24}$$

with $\hat{\mathbf{R}}_{\mathbf{z}_{f_\bullet}\boldsymbol{\xi}}(t) = \sum_{k=1}^{t} \beta^{t-k}\mathbf{z}_{f_\bullet}(k)\boldsymbol{\xi}^T(k)$. The minimization of such a cost function can therefore be realized by three different manners.

4.3.1 COIVPM, IVPM and EIVPM algorithms

COIVPM algorithm The first solution, named COIVPM, is similar to the COPM algorithm. Indeed, since the criterion (24) can be easily rewritten as

$$J_{IV}(\mathbf{P}_f(t)) = \sum_{i=1}^{n_\xi} \left\| \mathbf{r}^i_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) - \mathbf{P}_f^T(t)\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t) \right\|^2$$

where $\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}$ and $\mathbf{r}^i_{\mathbf{z}_{f_2}\boldsymbol{\xi}}$ are respectively the ith columns of the covariance matrices $\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}$ and $\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}$, it is possible to minimise the cost function (24) by sequentially minimising n_ξ criteria defined as

$$J_{IV_i}(\mathbf{P}_f(t)) = \sum_{k=1}^{t} \beta^{t-k} \left\| \mathbf{r}^i_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(k) - \mathbf{P}_f^T(t)\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(k) \right\|^2 \text{ for } i \in [1, n_\xi] \,.$$

By considering only vectors and not covariance matrices, the computational load of the global criterion optimization becomes lower. Each criterion J_{IV_i} can be minimised by a least squares algorithm similar to those developed for RPM1 and COPM. The following algorithm, named COIVPM, is summarised in Table 4.

```
Build $\bar{\bar{\mathbf{L}}}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \ldots$
    $\hat{\mathbf{R}}_{\mathbf{z}_f\boldsymbol{\xi}}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_f\boldsymbol{\xi}}(t-1) + \mathbf{z}_f(t)\boldsymbol{\xi}^T(t)$        $n_\xi n_y f$
    $i = \text{remainder}(t, n_\xi) + 1$
    $\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t) = \hat{\mathbf{R}}_{\mathbf{z}\boldsymbol{\xi}}(1 : n_x, i)(t)$
    $\mathbf{r}^i_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) = \hat{\mathbf{R}}_{\mathbf{z}\boldsymbol{\xi}}(n_x + 1 : n_y f, i)(t)$
    $\bar{\bar{\mathbf{K}}}_f(t) = \dfrac{\mathbf{r}^{i\,T}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t)\bar{\bar{\mathbf{L}}}_f(t-1)}{\beta + \mathbf{r}^{i\,T}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t)\mathbf{L}_f(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t)}$        $n_x(n_x + 1)$
    $\bar{\bar{\mathbf{L}}}_f(t) = \frac{1}{\beta}\left(\bar{\bar{\mathbf{L}}}_f(t-1) - \bar{\bar{\mathbf{L}}}_f(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t)\bar{\bar{\mathbf{K}}}_f(t)\right)$        $n_x^2$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{r}^i_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) - \mathbf{P}_f^T(t-1)\mathbf{r}^i_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t)\right)\bar{\bar{\mathbf{K}}}_f(t)$        $2n_x(n_y f - n_x)$
end
```

Table 4: The COIVPM algorithm and its computational load.

IVPM and EIVPM algorithms The minimization of the criterion (24) can be as well recursively realized more simply by considering the equation of its minimum. This one is differently written according to the dimension of the instrumental variable.

- If $n_\xi = n_x$, by assuming that the instrumental variable is enough correlated to the vector $\mathbf{z}_{f_1}$ such that the matrix $\mathbf{R}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}$ is invertible (see [25] for conditions on the system and the I/O data), the minimising argument of the cost function (24) is given by

$$\hat{\mathbf{P}}_f^T(t) = \hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}^{-1}(t). \tag{25}$$

- If $n_\xi > n_x$, the covariance matrix $\mathbf{R}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}$ is no more square. Its inversion needs the use of the Moore Penrose inverse [9]

$$\hat{\mathbf{P}}_f^T(t) = \hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}^{\dagger}(t).$$

Two algorithms are then developed in the following to estimate the propagator $\mathbf{P}_f$ according to the hypothesis on the instrumental variable dimension.

IVPM algorithm

If $n_\xi = n_x$, the estimation $\mathbf{P}_f$ is feasible by applying the matrix inversion lemma to the least squares solution (25). The following algorithm, named IVPM, is summarised in Table 5.

```
Build $\breve{\breve{\mathbf{L}}}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \dots$
    $\breve{\breve{\mathbf{K}}}_f(t) = \dfrac{\boldsymbol{\xi}^T(t)\breve{\breve{\mathbf{L}}}_f(t-1)}{\beta + \boldsymbol{\xi}^T(t)\breve{\breve{\mathbf{L}}}_f(t-1)\mathbf{z}_{f_1}(t)}$        $n_x(n_x+1)$
    $\breve{\breve{\mathbf{L}}}_f(t) = \dfrac{1}{\beta}\left(\breve{\breve{\mathbf{L}}}_f(t-1) - \breve{\breve{\mathbf{L}}}_f(t-1)\mathbf{z}_{f_1}(t)\breve{\breve{\mathbf{K}}}_f(t)\right)$        $n_x^2$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{z}_{f_2}(t) - \mathbf{P}_f^T(t-1)\mathbf{z}_{f_1}(t)\right)\breve{\breve{\mathbf{K}}}_f(t)$        $2n_x(n_y f - n_x)$
end
```

Table 5: The IVPM algorithm and its computational load.

EIVPM algorithm

When $n_\xi > n_x$, the use of the matrix inversion lemma is no more accurate since the matrix $\mathbf{R}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}$ is rectangular by construction. Now, this situation is not rare in practice since the instrumental variable is often built from delayed MIMO data which makes difficult to fix $n_\xi = n_x$ *a priori*. Furthermore, it was argued in [33] that the accuracy of the estimates obtained from an instrumental variable method increases with the number of instruments. In that case, the minimization of the criterion (24) requires the use of a technique named the Extended Instrumental Variable Method [8]. The application of such a method gives the recursive updating formulas introduced in Table 6.

```
Build $\mathbf{L}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \dots$
    $\mathbf{g}_f(t) = \begin{bmatrix}\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t-1)\boldsymbol{\xi}(t) & \mathbf{z}_{f_2}(t)\end{bmatrix}$        $n_\xi(n_y f - n_x)$
    $\boldsymbol{\Lambda}(t) = \begin{bmatrix}-\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t) & \beta \\ \beta & 0\end{bmatrix}$        $n_\xi$
    $\boldsymbol{\Psi}_f(t) = \begin{bmatrix}\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\boldsymbol{\xi}(t) & \mathbf{z}_{f_1}(t)\end{bmatrix}$        $n_\xi n_x$
    $\mathbf{K}_f(t) = \left(\boldsymbol{\Lambda}(t) + \boldsymbol{\Psi}_f^T(t)\mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t)\right)^{-1}\boldsymbol{\Psi}_f^T(t)\mathbf{L}_f(t-1)$        $2n_x(n_x+4)$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{g}_f(t) - \mathbf{P}_f^T(t-1)\boldsymbol{\Psi}_f(t)\right)\mathbf{K}_f(t)$        $4n_x(n_y f - n_x)$
    $\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1) + \mathbf{z}_{f_1}(t)\boldsymbol{\xi}^T(t)$        $n_\xi n_x$
    $\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t-1) + \mathbf{z}_{f_2}(t)\boldsymbol{\xi}^T(t)$        $n_\xi(n_y f - n_x)$
    $\mathbf{L}_f(t) = \dfrac{1}{\beta^2}\left(\mathbf{L}_f(t-1) - \mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t)\mathbf{K}_f(t)\right)$        $2n_x^2$
end
```

Table 6: The EIVPM algorithm and its computational load.

4.3.2 Improvement of the extended variable method: the EIVsqrtPM algorithm

As all recursive least squares algorithms, a necessary condition for the numerical stability of EIVPM is to impose that the matrix $\mathbf{L}_f$ is positive definite at each iteration. Indeed, when this matrix is ill-conditioned and not positive definite, the considered algorithm may diverge. Now, according to the update formula of $\mathbf{L}_f$ (see Table 6), this positiveness condition cannot be guaranteed. Indeed, even if the matrix $\mathbf{L}_f$ is positive definite at time $t-1$, the relation

$$\mathbf{L}_f(t) = \frac{1}{\beta^2}\left(\mathbf{L}_f(t-1) - \mathbf{L}_f(t-1)\mathbf{\Psi}_f(t)\left(\mathbf{\Lambda}(t) + \mathbf{\Psi}_f^T(t)\mathbf{L}_f(t-1)\mathbf{\Psi}_f(t)\right)^{-1}\mathbf{\Psi}_f^T(t)\mathbf{L}_f(t-1)\right)$$

does not verify this property at time t since

$$\mathbf{\Lambda}(t) + \mathbf{\Psi}_f^T(t)\mathbf{L}_f(t-1)\mathbf{\Psi}_f(t) = \begin{bmatrix} -\boldsymbol{\xi}^T(t)\left(\mathbf{I}-\hat{\mathbf{R}}^T_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\mathbf{L}_f(t-1)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\right)\boldsymbol{\xi}(t) & \beta+\boldsymbol{\xi}^T(t)\hat{\mathbf{R}}^T_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\mathbf{L}_f(t-1)\mathbf{z}_{f_1}(t) \\ \beta+\mathbf{z}^T_{f_1}(t)\mathbf{L}_f(t-1)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\boldsymbol{\xi}(t) & \mathbf{z}^T_{f_1}(t)\mathbf{L}_f(t-1)\mathbf{z}_{f_1}(t) \end{bmatrix} \tag{26}$$

has two eigenvalues of opposite[6] sign. Then, it is not possible to guarantee a good convergence of the EIVPM algorithm (or all methods using the EIV technique [33]). The first improvement (inspired by [11]) rests on the calculation of $\mathbf{L}_f$ as proposed in Table 6, the extraction of its upper triangular part and the use of its symmetry property to complete its update. Unfortunately, this approach cannot guarantee the positive definiteness of $\mathbf{L}_f$. The technique developed hereafter ensures this property. It consists more precisely in adapting an algorithm proposed by B. Porat et B. Friedlander [31] to the multivariable case. The underlying idea is to compute the square root of $\mathbf{L}_f$. This approach is a sufficient condition to preserve the positive definiteness of $\mathbf{L}_f$.

Assume that the matrix $\mathbf{L}_f$ is positive definite at time t. If $\mathbf{\Lambda}$ were positive definite, it would be possible to write $\mathbf{L}_f(t)$ as follows

$$\begin{aligned}\mathbf{L}_f(t) = \frac{1}{\beta^2}\Bigg(&\mathbf{L}_f^{1/2}(t-1)\mathbf{L}_f^{T/2}(t-1) - \mathbf{L}_f^{1/2}(t-1)\left(\mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1)\right)^T \\ &\left(\mathbf{\Lambda}^{1/2}(t)\mathbf{\Lambda}^{T/2}(t) + \mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1)\left(\mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1)\right)^T\right)^{-1} \\ &\mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1)\mathbf{L}_f^{T/2}(t-1)\Bigg).\end{aligned}$$

The update of this relation would be available by considering the following matrix

$$\mathbf{T}_f(t) = \begin{bmatrix} \mathbf{\Lambda}^{1/2}(t) & \mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1) \\ \mathbf{0} & \mathbf{L}_f^{1/2}(t-1) \end{bmatrix}$$

since

$$\mathbf{T}_f(t)\mathbf{T}_f^T(t) = \begin{bmatrix} \mathbf{\Lambda}(t) + \mathbf{\Psi}_f^T(t)\mathbf{L}_f(t-1)\mathbf{\Psi}_f(t) & \mathbf{\Psi}_f^T(t)\mathbf{L}_f(t-1) \\ \mathbf{L}_f(t-1)\mathbf{\Psi}_f(t) & \mathbf{L}_f(t-1) \end{bmatrix} \tag{27}$$

contains the components of $\mathbf{L}_f(t)$. Unfortunately, $\mathbf{\Lambda}$ (see Table 6) is not positive definite. Then, it is necessary to suggest an other way to update $\mathbf{L}_f(t)$. For that, notice that $\mathbf{\Lambda}$ can be rewritten as

$$\begin{aligned}\mathbf{\Lambda}(t) &= \begin{bmatrix} \sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)} & 0 \\ -\frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} & \frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)} & -\frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} \\ 0 & \frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} \end{bmatrix} \\ &= \mathbf{\Delta}(t)\mathbf{J}\mathbf{\Delta}^T(t)\end{aligned}$$

with $\mathbf{J}$ the signature matrix of $\mathbf{\Lambda}$ [9]. Thus, by defining

$$\bar{\mathbf{T}}_f(t) = \begin{bmatrix} \mathbf{\Delta}(t) & \mathbf{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1) \\ \mathbf{0} & {\mathbf{L}_f}^{1/2}(t-1) \end{bmatrix},$$

[6] note that $\mathbf{I} - \hat{\mathbf{R}}^T_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\mathbf{L}_f(t-1)\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)$ is a positive definite matrix.

Eq. (27) can be rewritten as

$$\bar{\mathbf{T}}_f(t)\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}\bar{\mathbf{T}}_f^T(t)=\begin{bmatrix}\boldsymbol{\Lambda}(t)+\boldsymbol{\Psi}_f^T(t)\mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t) & \boldsymbol{\Psi}_f^T(t)\mathbf{L}_f(t-1)\\ \mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t) & \mathbf{L}_f(t-1)\end{bmatrix}.$$

The next step consists therefore in updating the matrix $\bar{\mathbf{T}}_f(t)$. By referring to [9], we know that there is a matrix $\mathbf{Rot}_H \in \mathbb{R}^{2\times 2}$ composed by hyperbolic rotations such that $(\mathbf{Rot}_H)\mathbf{J}(\mathbf{Rot}_H)^T = \mathbf{J}$ and $\boldsymbol{\Delta}\mathbf{Rot}_H$ is lower triangular. Moreover [9], for all Givens rotation $\mathbf{Rot}_G$, $(\mathbf{Rot}_G)(\mathbf{Rot}_G)^T = \mathbf{I}$. The developed updating method adapted from [31] consists in associating both types of rotations into a single matrix named $\mathbf{Rot}$ in order to apply them to $\bar{\mathbf{T}}_f(t)$ so that their combined action leads to a triangular matrix

$$\bar{\mathbf{T}}_f(t)\mathbf{Rot}=\begin{bmatrix}\mathbf{L}_{11} & \mathbf{0}\\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}$$

with $\mathbf{L}_{11}$ and $\mathbf{L}_{22}$ two lower triangular matrices of respective dimension 2×2 and $n_x \times n_x$ and such that

$$\mathbf{Rot}\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}\mathbf{Rot}^T=\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}.$$

The links between these matrices and $\bar{\mathbf{T}}_f$ are displayed by noticing that

$$\begin{aligned}\bar{\mathbf{T}}_f(t)\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}\bar{\mathbf{T}}_f^T(t) &= \bar{\mathbf{T}}_f(t)\mathbf{Rot}\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}\mathbf{Rot}^T\bar{\mathbf{T}}_f^T(t)\\ &= \begin{bmatrix}\mathbf{L}_{11} & \mathbf{0}\\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{n_x}\end{bmatrix}\begin{bmatrix}\mathbf{L}_{11}^T & \mathbf{L}_{21}^T\\ \mathbf{0} & \mathbf{L}_{22}^T\end{bmatrix}.\end{aligned}$$

Then the following results hold

$$\boldsymbol{\Delta}(t)\mathbf{J}\boldsymbol{\Delta}^T(t)+\boldsymbol{\Psi}_f^T(t)\mathbf{L}_f(t)\boldsymbol{\Psi}_f(t)=\mathbf{L}_{11}\mathbf{J}\mathbf{L}_{11}^T, \tag{28}$$

$$\mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t)=\mathbf{L}_{21}\mathbf{J}\mathbf{L}_{11}^T, \tag{29}$$

$$\mathbf{L}_f(t-1)=\mathbf{L}_{21}\mathbf{J}\mathbf{L}_{21}^T+\mathbf{L}_{22}\mathbf{L}_{22}^T. \tag{30}$$

From Eq. (28) and (29), we have

$$\mathbf{K}_f(t)=\left(\mathbf{L}_{11}\mathbf{J}\mathbf{L}_{11}^T\right)^{-1}\boldsymbol{\Psi}_f^T(t)\mathbf{L}_f^T(t-1)=\left(\mathbf{L}_{11}\mathbf{J}\mathbf{L}_{11}^T\right)^{-1}\mathbf{L}_{11}\mathbf{J}\mathbf{L}_{21}^T=\mathbf{L}_{21}\mathbf{L}_{11}^{-1}. \tag{31}$$

From Eq. (29) we get

$$\mathbf{L}_{21}=\mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t)\mathbf{L}_{11}^{-T}\mathbf{J}^{-1}. \tag{32}$$

Finally, from (28), (31) and (32), Eq. (30) becomes

$$\mathbf{L}_{22}\mathbf{L}_{22}^T=\mathbf{L}_f(t-1)-\mathbf{L}_{21}\mathbf{J}\mathbf{L}_{21}^T=\mathbf{L}_f(t-1)-\mathbf{L}_f(t-1)\boldsymbol{\Psi}_f(t)\mathbf{K}_f(t)=\beta^2\mathbf{L}_f(t).$$

Hence,

$$\mathbf{L}_f^{1/2}(t)=\frac{1}{\beta}\mathbf{L}_{22}.$$

Furthermore, The full algorithm, named EIVsqrtPM, is summed up in Table 7.

4.4 Summary and comments

Seven possible implementations for recursive estimation of the propagator have been proposed. Although these techniques are conceptually similar, they have different characteristics and can be distinguished by their numerical properties and burdens. Table 8 introduces a summary of the number of multiplications necessary for each estimation phase. For information and comparison, the complexities of the main PAST-based methods [44, 11, 45] and of the eigendecomposition of the covariance matrix $\mathbf{R}_{\mathbf{z}_f}$ is given too. First of all, this table clearly indicates that all the developed recursive subspace identification methods have a low computational cost in comparison with the SVD one. Secondly, the comparison of the numerical load of the PM algorithms with the PAST by-products shows that all the techniques based on the propagator are computationally less demanding than their PAST equivalents. These complexity differences are mainly due to the facts that

```
Build $\mathbf{L}_f(0)$ and $\mathbf{P}_f^T(0)$ suitably, e.g. from parts of the ident. mat.
for $t = 1, \dots$
    $\mathbf{g}_f(t) = \begin{bmatrix} \hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t-1)\boldsymbol{\xi}(t) & \mathbf{z}_{f_2}(t) \end{bmatrix}$    $n_\xi(n_y f - n_x)$
    $\boldsymbol{\Delta}(t) = \begin{bmatrix} \sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)} & 0 \\ -\frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} & \frac{\beta}{\sqrt{\boldsymbol{\xi}^T(t)\boldsymbol{\xi}(t)}} \end{bmatrix}$    $n_\xi$
    $\boldsymbol{\Psi}_f(t) = \begin{bmatrix} \hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1)\boldsymbol{\xi}(t) & \mathbf{z}_{f_1}(t) \end{bmatrix}$    $n_\xi n_x$
    Find the rotations matrix Rot such that
    $\begin{bmatrix} \boldsymbol{\Delta}(t) & \boldsymbol{\Psi}_f^T(t)\mathbf{L}_f^{1/2}(t-1) \\ \mathbf{0} & \mathbf{L}_f^{1/2}(t-1) \end{bmatrix} \mathbf{Rot} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}$    $2n_x(n_x+3)^3 + 8n_x$
    $\mathbf{L}_f^{1/2}(t) = \frac{1}{\beta}\mathbf{L}_{22}$
    $\mathbf{K}_f(t) = \left(\mathbf{L}_{21}\mathbf{L}_{11}^{-1}\right)^T$    $4n_x$
    $\mathbf{P}_f^T(t) = \mathbf{P}_f^T(t-1) + \left(\mathbf{g}_f(t) - \mathbf{P}_f^T(t-1)\boldsymbol{\Psi}_f(t)\right)\mathbf{K}_f(t)$    $4n_x(n_y f - n_x)$
    $\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_{f_1}\boldsymbol{\xi}}(t-1) + \mathbf{z}_{f_1}(t)\boldsymbol{\xi}^T(t)$    $n_\xi n_x$
    $\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t) = \lambda\hat{\mathbf{R}}_{\mathbf{z}_{f_2}\boldsymbol{\xi}}(t-1) + \mathbf{z}_{f_2}(t)\boldsymbol{\xi}^T(t)$    $n_\xi(n_y f - n_x)$
end
```

Table 7: The EIVsqrtPM algorithm and its computational load.

- a computation step is added in the PAST methods (relatively to the PM techniques) to perform the projection approximation of the Yang's criterion,
- the PM approaches only estimate the propagator, the dimension of which is less than the observability matrix one.

In order to simplify the analysis of the complexity of the proposed algorithms, a classic practical case is considered: $n_\xi = n_u p$ with $p = f$ which corresponds to choose the instruments as past input data

$$\boldsymbol{\xi}(t) = \begin{bmatrix} \mathbf{u}^T(t-p) & \cdots & \mathbf{u}^T(t-1) \end{bmatrix}^T. \tag{33}$$

Then, as shown in Table 9, the computational load of the PI/PO RQ factorization update is about twice as high than the one of the ordinary RQ factorization. Now, since the different propagator estimation methods, except EIVsqrtPM, have a complexity of the same order, the cheapest techniques from a computational point of view are those which use the ordinary RQ factorization update. Concerning EIVsqrtPM, the numerical stability improvement is payed by a significant complexity increase. The choice between EIVPM and EIVsqrtPM has to be done according to the well conditioning and the excitation quality of the experiment conditions.

In order to help the user for the choice of the suitable algorithms according to the application scenario, the main practical conditions of the developed propagator-based algorithms are introduced in Table 10. This table shows that

- RPM1 can only be used with noise-free data,
- IVPM should be applied to SISO systems so as to render easier the construction of the instrumental variable,
- RPM2 and COPM should be employed for systems with relatively small values of n_x, n_u and n_y,
- COIVPM and EIVPM can be used in most practical cases,
- EIVsqrtPM should be mainly used when the experimental conditions or the identified system features lead to numerical problems.

The use of the propagator for recursive subspace identification introduces a particular property for the model structure: the setting of the state-space basis. Indeed, if the system matrices $\mathbf{A}$ and $\mathbf{C}$ are extracted as in the batch case *i.e.* from $\hat{\boldsymbol{\Gamma}}_f$

$$\hat{\mathbf{C}} = \hat{\boldsymbol{\Gamma}}_f(1 : n_y, :),$$

$$\hat{\mathbf{A}} = \left\{\hat{\boldsymbol{\Gamma}}_f(1 : n_y(f-1), :)\right\}^\dagger \hat{\boldsymbol{\Gamma}}_f(n_y + 1 : n_y f, :),$$

Algorithm	RQ update compl. $\mathcal{O}(\bullet)$	$\mathbf{\Gamma}_f$ estimation compl. $\mathcal{O}(\bullet)$	Approx.	Bias
RPM1	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f n_x$	No	Yes/Yes
RPM2	$\left(\frac{n_u f+n_\xi}{2}+n_y f\right)(n_u f+n_\xi)$	$2n_x(2n_y f+n_x)$	No	No/No
COPM	$\left(\frac{n_u f+n_\xi}{2}+n_y f\right)(n_u f+n_\xi)$	$2n_y f(2n_y f+n_x)$	No	No/No
COIVPM	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$n_y f(n_\xi+2n_x)$	No	No/No
IVPM	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f n_x$	No	No/No
EIVPM	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f(n_\xi+2n_x)$	No	No/No
EIVsqrtPM	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f(n_\xi+2n_x)+2n_x(n_x+3)^3$	No	No/No
PAST	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$3n_y f n_x+2n_x^2$	Yes	No/Yes
COPAST	$\left(\frac{n_u f+n_\xi}{2}+n_y f\right)(n_u f+n_\xi)$	$n_y f(2n_y f+3n_x)+2n_x^2$	Yes	No/No
EIVPAST	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2(n_y f+n_x)(n_\xi+2n_x)$	Yes	No/No
ED or SVD	$\left(\frac{n_u f+n_\xi}{2}+n_y f\right)(n_u f+n_\xi)$	$n_y^3 f^3$	No	No/No

Table 8: Major computational load of the main recursive subspace algorithms. ”Approx.” indicates if the algorithm introduces an approximation for the $\mathbf{\Gamma}_f$ estimation. ”Bias” indicates if the algorithm gives access to a biased estimate of $\mathbf{\Gamma}_f$ when $\mathbf{b}_f$ is a white/coloured Gaussian noise.

Algorithm	RQ update meth.	RQ update compl. $\mathcal{O}(\bullet)$	$\mathbf{\Gamma}_f$ estimation compl. $\mathcal{O}(\bullet)$
RPM1	Ord. MOESP	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f n_x$
RPM2	PI/PO MOESP	$2(n_u+n_y)n_u f^2$	$2n_x(2n_y f+n_x)$
COPM	PI/PO MOESP	$2(n_u+n_y)n_u f^2$	$2n_y f(2n_y f+n_x)$
COIVPM	Ord. MOESP	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$n_y f(n_u f+2n_x)$
IVPM	Ord. MOESP	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f n_x$
EIVPM	Ord. MOESP	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f(n_u f+2n_x)$
EIVsqrtPM	Ord. MOESP	$\left(\frac{n_u}{2}+n_y\right)n_u f^2$	$2n_y f(n_u f+2n_x)+2n_x(n_x+3)^3$

Table 9: Major computational load of the developed propagator-based algorithms when $n_\xi=n_u p$ with $p=f$.

Algorithm	Noisy data	MIMO system	Ill-conditioned mat.	High values of n_x, n_u and n_y
RPM1		✓		✓
RPM2	✓	✓		
COPM	✓	✓		
COIVPM	✓	✓		✓
IVPM	✓			✓
EIVPM	✓	✓		✓
EIVsqrtPM	✓	✓	✓	

Table 10: Main application conditions of the developed propagator-based algorithms.

the parametric structure of the state-space matrices is frozen throughout the identification procedure since, in our case, $\hat{\mathbf{\Gamma}}_f = \begin{bmatrix}\mathbf{I}_{n_x} & \hat{\mathbf{P}}_f\end{bmatrix}^T$. This loss of flexibility can be considered as a drawback, more particularly from a numerical point of view. Indeed, the fact that the first n_x rows of $\hat{\mathbf{\Gamma}}_f$ are equal to the identity matrix could lead to conditioning problems when the proposed methods are applied to ill-conditioned systems. However, we have not encountered any numerical problems when simulating the algorithms in MATLAB®. Furthermore, contrary to the PAST approach where the signal subspace is obtained by minimising a modified function obtained after an approximation, the PM cost functions are always determined without approximation nor constraint. This is an interesting property of the propagator algorithms, in particular when RSMI techniques are used for change detection purposes.

5 Simulation examples

5.1 MIMO time-invariant and time-varying systems

In order to illustrate the performances of the recursive algorithms proposed in this paper, the developed methods are applied to the I/O data of the following state-space system used , *e.g.*, by H. Oku in [28, Chapter4]

$$\mathbf{x}(t+1) = \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.3 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ 0 & -0.6 \\ 0.5 & 0 \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} 0.055 \\ 0.04 \\ 0.045 \end{bmatrix} w(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0.025 \\ 0.03 \end{bmatrix} v(t).$$

Three different practical situations are more precisely considered: a time-invariant case, a slowly time-varying case and a time-varying with abrupt change case. For each case, the I/O data are generated by following the same experimental procedure:

- The initial state-space matrices are randomly generated under the constraint that the absolute value of the maximum eigenvalue of $\hat{\mathbf{A}}(0)$ is less than 1 (stability requirement).
- A Monte Carlo simulation (MCS) of size 100 is carried out.
- The input and noise signals are zero mean white Gaussian noises of variance 1 such that the signal to noise ratio[7] (SNR) equals $25dB$.

The instrumental variable is chosen as past input data (see Eq. (33)) and the hyper parameters f and p are fixed equal to 5. The forgetting factors are built as follows

$$\lambda(t) = \beta(t) = \min\left\{\lambda_0\lambda(t-1) + 1 - \lambda_0, \lambda_{final}\right\}$$

in order to quickly annihilate the effects of this initialization.

Remark 6 *As far as the choice of the tuning parameters p and f is concerned, the user is often faced with a tricky situation. The study of several simulation examples with systems of order $n_x < 10$ has displayed that choosing $f = p = n_x + 2$ gives results as good as those obtained with $f = p = 3n_x$ or more. This observation is an interesting characteristic since, as shown in Table 8, the computational load of the RSMI methods strongly depends on these values. Choosing them relatively small is also interesting from a numerical point of view.*

5.1.1 Time-invariant case

Firstly, the performances of the developed online algorithms are examined with the help of time-invariant simulations. For that, the subspace angles (in radians) between the average estimated and true observability matrices is used as measure of quality. This approach provides indeed an easy plotting scalar

[7] The signal to noise ratio is defined as $SNR = 20\log\left(\frac{\text{std}\{\mathbf{y}_d\}}{\text{std}\{\mathbf{v}\}}\right)$ where $\mathbf{y}_d$ is the deterministic part of $\mathbf{y}$.

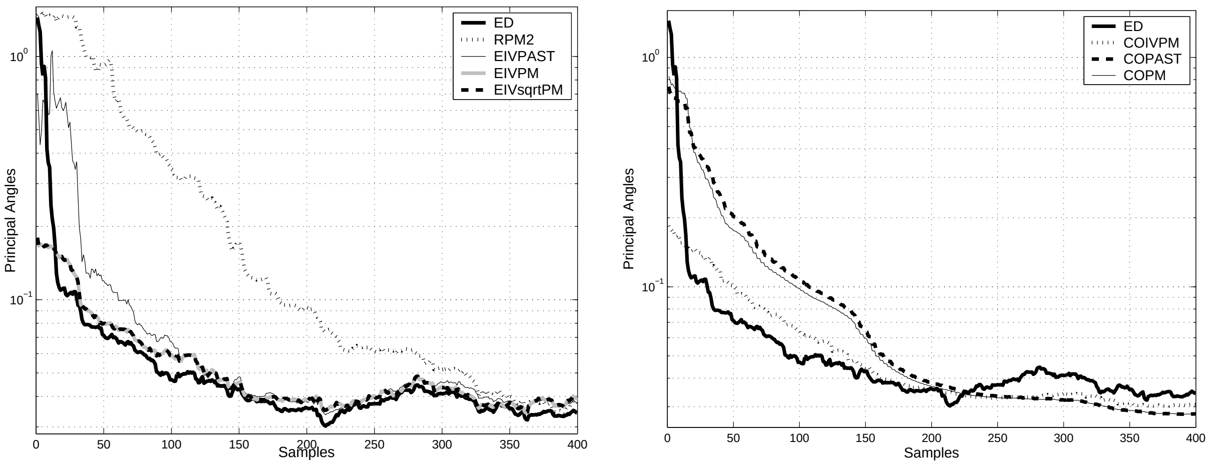

Figure 1: Comparison of the subspace angles computed with RPM2, EIVPM, EIVsqrtPM, COPM, COIVPM, EIVPAST and COPAST in a stationary environment. $\lambda_0 = 0.98$, $\lambda(0) = 0.98$, $\lambda_{final} = 0.999$. $SNR = 25dB$, MCS=100.

quantity suitable for the transient effects exam. In order to demonstrate the efficiency of the recursive subspace identification methods, an eigendecomposition-based estimate obtained from the n_x principal eigenvectors of $\hat{\mathbf{R}}_{\mathbf{z}_f}$ is also added in the study (ED on Figure 1).

Remark 7 *For clarity, the plots obtained with RPM1 and IVPM are not displayed on Figure 1. This choice can be justified by the fact that RPM1 leads to biased estimate in a noisy framework (see §4.1.4) and IVPM requires some constraints to build an instrumental variable of good size. Concerning the comparison of the PM and PAST approaches, it is important to notice that, originally, COPAST was developed for array signal processing. In this paper, it has been necessary to adapt this method for the observability matrix subspace tracking. To the authors' knowledge, there is no COIVPAST algorithm in the literature.*

As can be seen on Figure 1, all the propagator-based methods provide consistent estimates of the observability subspace, since, as time goes on, the principal angles between $\mathbf{\Gamma}_f$ and $\hat{\mathbf{\Gamma}}_f$ tends to 0. Although all the estimates asymptotically converge to the true subspace, the convergence speed and the transient behaviour strongly differ according to the considered algorithm. Thus, the RPM2 average estimate is much slower to reach the system observability subspace than the other techniques. This algorithm is indeed based on two interlinked loops and uses block matrices such as $\underline{\mathbf{L}}_f$ or $\underline{\underline{\mathbf{L}}}_f$ (see Table 2) which decrease the forgetting ability of the developed method. This disadvantage could be reduced by considering smaller forgetting factors. Furthermore, we can notice that the subspace angles obtained with COPM and COIVPM have a smoother decrease in comparison with the other algorithms. This feature may be due to the sequential procedure used by both techniques. Finally, for this good-conditioned process, EIVPM and EIVsqrtPM leads to the same subspace estimates as shown on the left-hand side curves of Figure 1. This illustrates the fact that EIVsqrtPM is only a more numerically robust version of EIVPM. Concerning the PM and PAST comparison, the EIVPM and COPM algorithms show better performances in terms of initial convergence speed with respect to EIVPAST and COPAST. The slower convergence of PAST versions is most likely due to the projection approximation introduced to render easier the minimization of the Yang's criterion [44]. Thus, based on these simulation results, the EIVPM and COIVPM algorithms can be considered as the best solution for this time-invariant scenario.

5.1.2 Slowly time-varying case

In order to investigate the tracking performance of the developed recursive algorithm, a second experimental framework is considered. More precisely, after a time-invariant phase of size 665, the following

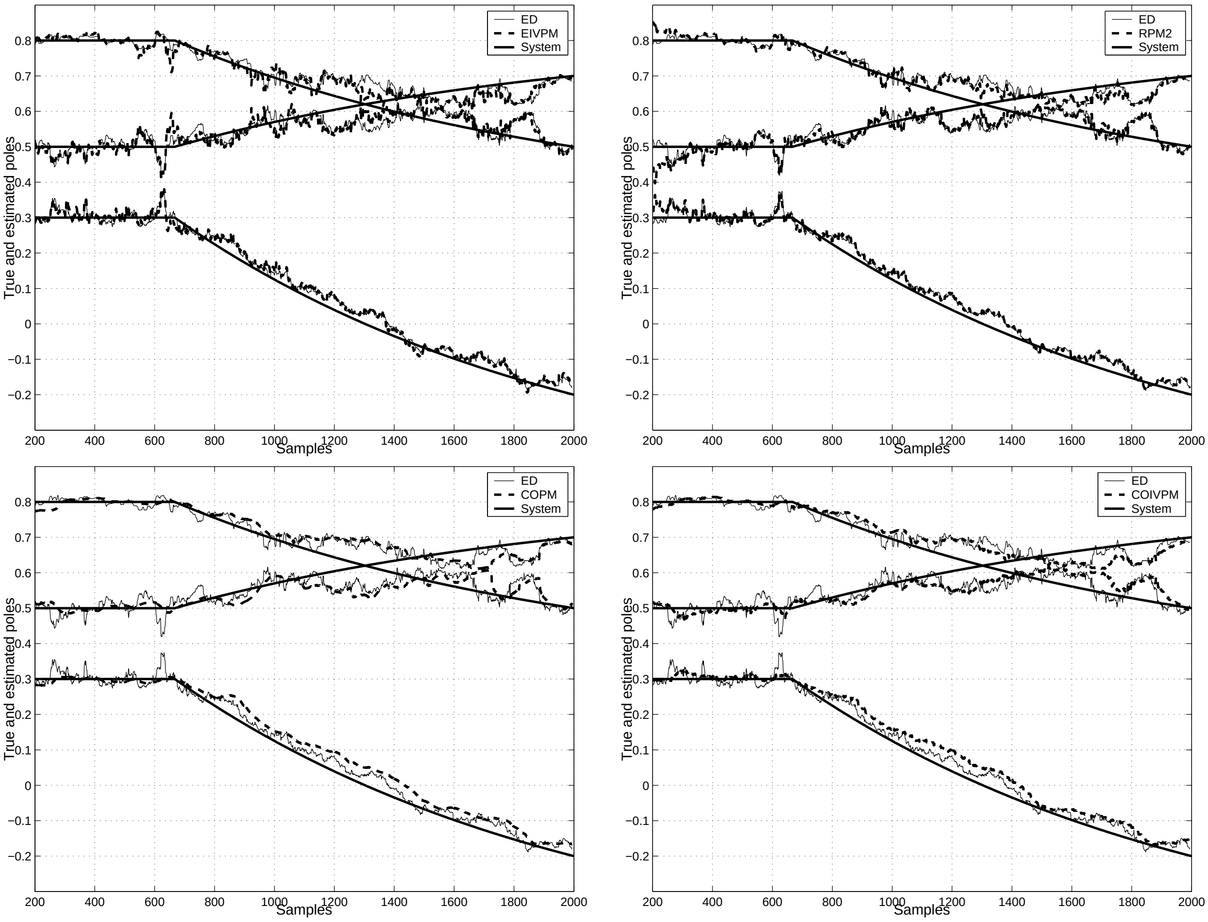

Figure 2: Poles trajectories estimated with EIVPM, RPM2, COPM COIVPM and ED in a nonstationary environment. $\lambda_0 = 0.97$, $\lambda(0) = 0.97$, $\lambda_{final} = 0.98$. $SNR = 25dB$, MCS=100.

state matrix is used

$$\mathbf{A} = \begin{bmatrix} 0.8 - 0.3\frac{\exp(-\frac{t-665}{2000})-1}{\exp(-1)-1} & -0.4 & 0.2 \\ 0 & 0.3 - 0.5\frac{\exp(-\frac{t-665}{2000})-1}{\exp(-1)-1} & -0.5 \\ 0 & 0 & 0.5 + 0.2\frac{\exp(-\frac{t-665}{2000})-1}{\exp(-1)-1} \end{bmatrix}.$$

Thus, the poles of the the system drift from {0.3, 0.5, 0.8} to {-0.2, 0.7, 0.5} during the next 1335 samples. The same conditions of experiment as in the previous Subsection are used, except for the forgetting factor which is set at $\lambda_0 = 0.97$, $\lambda(0) = 0.97$, $\lambda_{final} = 0.98$. The main characteristics of the system are the slowly time-varying of the system dynamic and the intersection of two eigenvalues. The estimated poles trajectories averaged over the 100 MCS runs are displayed on Figure 2. These curves show that all the techniques present interesting abilities to track variations in the system parameters. We can furthermore notice that EIVPM and RPM2 have performance very similar to the eigendecomposition-based approach (named ED on Figure 2). This observation proves that a significant complexity reduction can be achieved without scarifying accuracy. The trajectories obtained with COIVPM and COPM are smoother and relatively slower than with the other techniques. This property is in all likelihood due to the sequentially minimization procedure of the COPM and COIVPM algorithms. The use of such an approach leads indeed to an averaging of the estimates. Concerning the intersection scenario, EIVPM, RPM2 and ED know the same difficulties to discriminate the eigenvalues. In front of these results, using COPM or COIVPM for slowly time-varying system identification can be seen as a good solution since these techniques lead to smooth responses and offer a good trade-off between tracking and stationary performance. It is however important to notice that COIVPM is more interesting than COPM from a numerical complexity point of view. The choice between both methods can be done according to the computation time constraints.

5.1.3 Time-varying with abrupt change case

Although some assumptions necessary to the implementation of the recursive subspace identification algorithms are no more verified when a sudden and abrupt change is considered, the goal of the last simulation example is to study the behaviour of the introduced RSMI methods in this framework. For that, the following state matrix is used

$$\mathbf{A} = \begin{cases} \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.3 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix} & \text{for } t < 665 \\ \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.45 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix} & \text{for } t \geq 665 \end{cases}.$$

All the studied techniques are able to detect and track the pole trajectory jump. As previously (see Fig. 3), the EIVPM and RPM2 algorithms provide estimates analogous to those of the eigendecomposition-based approach. Their adaptation to this step change is faster than with COPM or COIVPM. However, this quickness is paid at the cost of a stronger variance. Once again, a trade-off between good tracking and smoothing response is thereby necessary.

5.2 Influence of the S matrix

In the previous example, the permutation matrix $\mathbf{S}$ introduced in Subsection 4.1.2 has been set to the identity since, in this particular case, the first n_x rows of the extended observability are linearly independent. Unfortunately, in many situations, choosing the $\mathbf{S}$ matrix in such way may result in a wrong partition of $\mathbf{\Gamma}_f$ and bad performances of the identification algorithms. The goal of following simulation example is to emphasise the need of well choosing this partitioning matrix. The example is intentionally

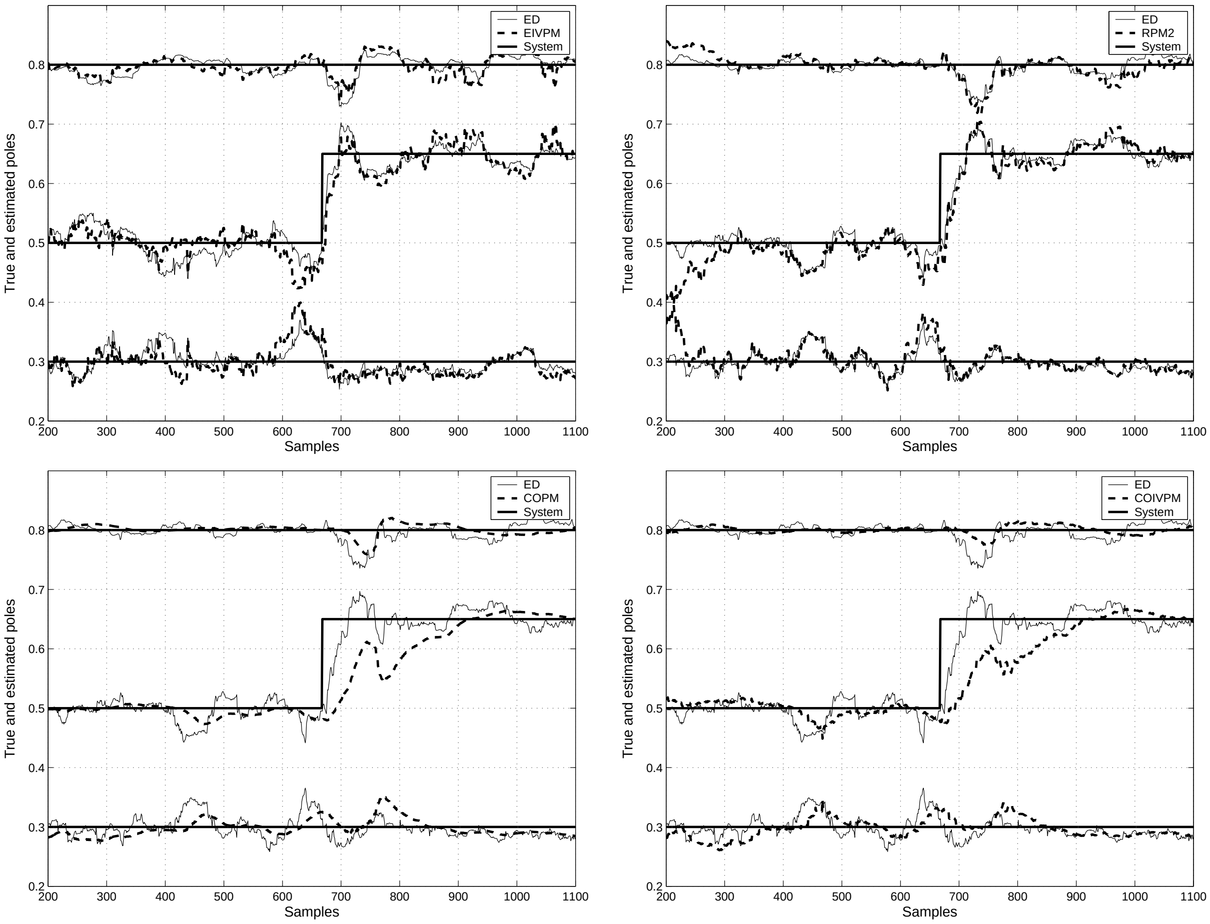

Figure 3: Poles trajectories estimated with EIVPM, RPM2, COPM COIVPM and ED in an abrupt nonstationary environment. $\lambda_0 = 0.97$, $\lambda(0) = 0.97$, $\lambda_{final} = 0.98$. $SNR = 25dB$, MCS=100.

set such that the first n_x rows of $\mathbf{\Gamma}_f$ are not linearly independent. It consists more particularly in

$$\mathbf{x}(t+1) = \begin{bmatrix} -0.850 & 0 & 0 \\ 0 & 0 & -1.4744 \\ 0 & 0.50 & 1.730 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} -0.70 & -0.30 \\ -0.4652 & -0.5972 \\ 1.4050 & 1.270 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 0.50 & 0 & 0 \\ 0 & 0 & 2.0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{u}(t) + v(t)$$

with $\mathbf{u}$ and v two zero mean white Gaussian noises such that SNR=$25dB$. As previously, a Monte Carlo simulation of size 100 is driven using respectively the EIVPM[8] with and without permutations, EIVPAST and a classic eigendecomposition. For EIVPM, the matrix $\mathbf{S}$ is selected by following the trick suggested in Subsection 4.1.2. The average principal angles between the estimated observability space and the true one are depicted on Figure 4. These plots firstly illustrate the necessity of rearranging the rows of $\mathbf{\Gamma}_f$ for improving the efficiency of EIVPM. EIVPM with $\mathbf{S}$ introduction leads indeed to accurate $\mathbf{\Gamma}_f$ estimates in less than 300 samples. On the contrary, the principal angle computed with EIVPM without $\mathbf{S}$ application has a chaotic dynamic which will result in bad state-space matrices estimation. Secondly, it is interesting to notice that, even if the EIVPAST estimates have better asymptotic behaviour than the "EIVPM without permutations" ones, the EIVPAST estimates converges less quickly to the true observability subspace than when the $\mathbf{S}$ matrix is used.

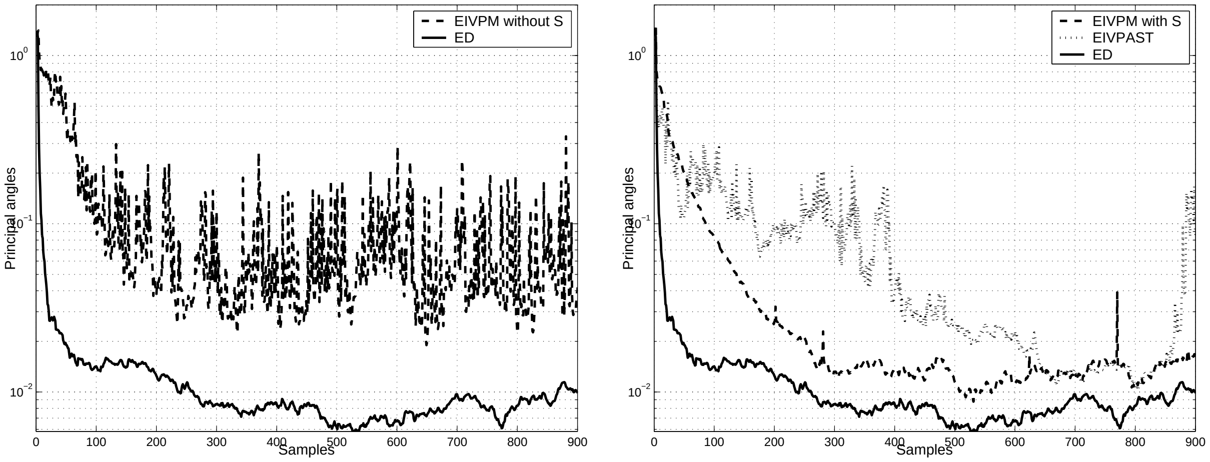

Figure 4: Comparison of the subspace angles computed with EIVPM with and without the use of the $\mathbf{S}$ matrix and EIVPAST. $\lambda_0 = 0.98$, $\lambda(0) = 0.98$, $\lambda_{final} = 0.999$. $SNR = 25dB$, MCS=100.

6 Conclusion

In this paper, several recursive subspace model identification algorithms are proposed. The estimation of the observability subspace of MIMO systems from I/O data is more precisely formulated as specific unconstrained optimization problem adapted from signal processing. This adaptation, combined with the introduction of particular instruments, allow to avoid the use of singular values decomposition to recursively estimate the signal subspace in a noisy framework. A particular attention is paid to the computational load of the developed methods. The performances of these algorithms are highlighted with some simulation examples. The experimental results indicate more particularly that propagator-based approach has faster initial convergence speed than their PAST counterparts and that the proposed techniques have good behaviour and performances when slowly time-varying systems and sudden changes in the process dynamic are considered. Thus, the developed algorithms provide a satisfactory balance between computational complexity and estimation accuracy.

[8] The same results are obtained with all the propagator-based algorithms.

References

[1] M. Abdelghani, M. Verhaegen, P. Van Overschee, and B. De Moor. Comparison study of subspace identification methods applied to flexible structures. *Mechanical Systems and Signal Processing*, 12:679–692, 1998.

[2] D. Bauer. Asymptotic properties of subspace estimators. *Automatica*, 41:359–376, 2005.

[3] G. Bienvenu and L. Kopp. Optimality of high resolution array processing using the eigensystem approach. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31:1235–1247, 1983.

[4] Y. M. Cho, G. Xu, and T. Kailath. Fast recursive identification of state space models via exploitation displacement structure. *Automatica*, 30:45–60, 1994.

[5] C. T. Chou and M. Verhaegen. Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica*, 33:1857–1869, 1997.

[6] K. De Cock, Guillaume Mercère, and B. De Moor. Recursive subspace identification for in flight modal analysis of airplanes. In *Proceedings of the International Conference on Noise and Vibration Engineering*, Leuven, Belgium, September 2006.

[7] W. Favoreel, B. De Moor, and P. Van Overschee. Subspace state space system identification for industrial processes. *Journal of Process Control*, 10:149–155, 2000.

[8] B. Friedlander. The overdetermined recursive instrumental variable method. *IEEE Transactions on Automatic Control*, 4:353–356, 1984.

[9] G. H. Golub and C. F. Van Loan. *Matrix computations*. John Hopkins University Press, Baltimore MD, 3rd edition, 1996.

[10] T. Gustafsson. Recursive system identification using instrumental variable subspace tracking. In *Proceedings of the 11th IFAC Symposium on System Identification*, Fukuoka, Japan, September 1997.

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

[12] K. Kameyama, A. Oshumi, Y. Matsuüra, and K. Sawada. Recursive 4SID based identification algorithm with fixed input output data size. *International Journal of Innovative Computing Information and Control*, 1:17–33, 2005.

[13] T. Katayama. *Subspace methods for system identification*. Springer, 2005.

[14] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13:67–94, 1996.

[15] S. Kukreja, B Haverkamp, D. Westwick, R. Kearney, H. Galiana, and M. Verhaegen. Subspace identification method for ankle mechanics. *IEEE Engineering in Medecine and Biology Society*, 17:1413–1414, 1995.

[16] W. E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, Hawaii, USA, December 1990.

[17] L. Ljung. *System identification. Theory for the user.* PTR Prentice Hall Information and System Sciences Series. T. Kailath, Series Editor, Upper Saddle River, 2nd edition, 1999.

[18] M. Lovera. *Subspace identification methods: theory and applications.* PhD thesis, Dipartimento di Elettronica e Informazione, Politechnico di Milano, 1997.

[19] M. Lovera. Recursive subspace identification based on projector tracking. In *Proceedings of the 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, August 2003.

[20] M. Lovera, T. Gustafsson, and M. Verhaegen. Recursive subspace identification of linear and non linear Wiener state space models. *Automatica*, 36:1639–1650, 2000.

[21] G. Mercère, S. Lecœuche, and M. Lovera. Recursive subspace identification based on instrumental variable unconstrained quadratic optimization. *International Journal of Adaptive Control and Signal Processing*, 18:771–797, 2004.

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

[23] G. Mercère, S. Lecœuche, and C. Vasseur. Output noise insensitive identification of systems: a new recursive state space method. In *Proceedings of the 3rd Workshop on Physics in Signal and Image Processing*, Grenoble, France, January 2003.

[24] G. Mercère, S. Lecœuche, and C. Vasseur. Sequential correlation based propagator algorithm for recursive subspace identification. In *Proceedings of the 16th IFAC World Congress*, Praha, Czech Republic, July 2005.

[25] G. Mercère and M. Lovera. Convergence analysis of instrumental variable recursive subspace identification algorithms. *Automatica*, 43:1377–1386, 2007.

[26] L. Mevel, L. Hermans, and H. Van der Auweraer. On the application of subspace-based fault detection methods to industrial structures. *Mechanical Systems and Signal Processing*, 13:823–838, 1999.

[27] J. Munier and G. Y. Delisle. Spatial analysis using new properties of the cross spectral matrix. *IEEE Transactions on Signal Processing*, 39:746–749, 1991.

[28] H. Oku. *Sequential subspace state space system identification and state estimation of unknown multivariable systems.* PhD thesis, Department of Mathematical Engineering, Information and Physics, University of Tokyo, Tokyo, Japan, 2000.

[29] H. Oku and H. Kimura. A recursive 4SID from the input output point of view. *Asian Journal of control*, 1:258–269, 1999.

[30] H. Oku and H. Kimura. Recursive 4SID algorithms using gradient type subspace tracking. *Automatica*, 38:1035–1043, 2002.

[31] B. Porat and B. Friedlander. The square root overdetermined recursive instrumental variable algorithm. *IEEE Transactions on Automatic Control*, 34:656–658, 1989.

[32] R. Roy, A. Paulraj, and T. Kailath. ESPRIT: a subspace rotation approach to estimation of parameters of cisoids in noise. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34:1340–1342, 1986.

[33] T. Söderström and P. Stoica. *System identification.* Prentice Hall International Series in Systems and Control Engineering, New York, 1989.

[34] R. J. Vaccaro. A second order perturbation expansion for the SVD. *SIAM Journal on Matrix Analysis and Applications*, 15:661–671, 1994.

[35] S. Van Huffel and J. Vandervalle. *The total least squares problem: computational aspects and analysis, frontiers in applied mathematics series*, volume 9. SIAM, Philadelphia, 1991.

[36] P. Van Overschee and B. De Moor. N4SID: subspace algorithms for the identification of combined deterministic stochastic systems. *Automatica*, 30:75–93, 1994.

[37] M. Verhaegen. Subspace model identification part 3: analysis of the ordinary output error state space model identification algorithm. *International Journal of Control*, 58:555–586, 1993.

[38] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input output data. *Automatica*, 30:61–74, 1994.

[39] M. Verhaegen and E. Deprettere. A fast recursive MIMO state space model identification algorithm. In *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, United Kingdom, December 1991.

[40] M. Verhaegen and P. Dewilde. Subspace model identification part 1: output error state space model identification class of algorithms. *International Journal of Control*, 56:1187–1210, 1992.

[41] M. Verhaegen and P. Dewilde. Subspace model identification part 2: analysis of the elementary output error state space model identification algorithm. *International Journal of Control*, 56:1211–1241, 1992.

[42] M. Verhaegen, A. Varga, and G. Grübel. Some experience with the MOESP class of subspace model identification methods in identifying the BO105 helicopter. Technical report, German Aerospace Research Establishment, Laboratory for Flight Systems Dynamics, Wessling, Germany, 1994.

[43] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31:1835–1851, 1995.

[44] B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43:95–107, 1995.

[45] J. L. Yu. A novel subspace tracking using correlation based projection approximation. *Signal Processing*, 80:2517–2525, 2000.