YET ANOTHER SUBSPACE TRACKER

Roland BADEAU, Bertrand DAVID, Gaël RICHARD

Télécom Paris - Département TSI
46 rue Barrault - 75634 PARIS cedex 13

ABSTRACT

This paper introduces a new algorithm for tracking the major subspace of the correlation matrix associated with time series. This algorithm greatly outperforms many well-known subspace trackers in terms of subspace estimation. Moreover, it guarantees the orthonormality of the subspace weighting matrix at each iteration, and reaches the lowest complexity found in the literature.

1. INTRODUCTION

Subspace tracking has been widely investigated in the fields of adaptive filtering, source localization or parameter estimation. Recently, a new subspace tracker, referred to as the Subspace Projection (SP) algorithm was proposed by C.E. Davila [1]. We observed in [2] that this algorithm greatly outperforms many well-known subspace trackers in terms of subspace estimation, such as Karasalo's algorithm [3], PAST [4], Loraf [5], FST [6], NIC [7], and OPAST [8]. As in [4] and [7], the estimation of the signal subspace is viewed as an optimization problem. However, instead of introducing approximations, the SP algorithm computes the signal subspace as the exact solution of this problem over a subspace of limited dimension.

Unfortunately, this remarkable subspace tracker is only suitable for time series data analysis, and has a high computational cost. Indeed, the global complexity of the SP algorithm is $O(nr^2)$ (where n is the dimension of the observed data vectors, and r is the dimension of the signal subspace), whereas a number of existing subspace trackers only require $O(nr)$ operations per time step (this is the case of PAST, FST, NIC and OPAST). Nevertheless, we noticed that this second drawback can be circumvented, and we propose in this paper a new algorithm, referred to as YAST, which computes the same signal subspace as the SP algorithm, but only requires $O(nr)$ operations.

The paper is organized as follows. In section 2, the basic principle of the YAST algorithm is presented. Then a fast implementation of YAST is proposed in section 3. The performance of this subspace tracker is illustrated in section 4. Finally, the main conclusions of this paper are summarized in section 5.

2. PRINCIPLE

Let $\{\boldsymbol{x}(t)\}_{t\in\mathbb{Z}}$ be a sequence of n-dimensional data vectors. We are interested in tracking the major subspace spanned by its correlation matrix $\boldsymbol{C}_{xx}(t)$. This matrix can be recursively updated according to

$$\boldsymbol{C}_{xx}(t) = \beta\,\boldsymbol{C}_{xx}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{x}(t)^H \tag{1}$$

where $0 < \beta < 1$ is the forgetting factor.

The YAST algorithm relies on the following principle: a $n \times r$ orthonormal matrix $\boldsymbol{W}(t)$ spans the r-dimensional major subspace of $\boldsymbol{C}_{xx}(t)$ if and only if it maximizes the criterion

$$\mathcal{J}\left(\boldsymbol{W}(t)\right) = \operatorname{trace}\left(\boldsymbol{W}(t)^H\boldsymbol{C}_{xx}(t)\boldsymbol{W}(t)\right).$$

In particular, the maximum of this criterion is equal to the sum of the r highest eigenvalues of $\boldsymbol{C}_{xx}(t)$. However, implementing this maximization over all orthonormal matrices is computationally demanding (the complexity is $O(n^2r)$), and does not lead to a simple recursion between $\boldsymbol{W}(t)$ and $\boldsymbol{W}(t-1)$.

In order to reduce the computational cost, the idea introduced in [1] consists in limiting this search to the range space of $\boldsymbol{W}(t-1)$ plus one or two additional search directions. In other words, the r-dimensional range space of $\boldsymbol{W}(t)$ is to be found as a subspace of the $(r+p)$-dimensional space spanned by the $n\times(r+p)$ matrix

$$\underline{\boldsymbol{V}}(t) = [\boldsymbol{W}(t-1),\, \underline{\boldsymbol{v}}(t)] \tag{2}$$

where $\underline{\boldsymbol{v}}(t)$ contains $p = 1$ or 2 columns. In practice, it is proposed in [1] to chose $\underline{\boldsymbol{v}}(t) = \boldsymbol{x}(t)$ or $\underline{\boldsymbol{v}}(t) = [\boldsymbol{x}(t),\, \boldsymbol{C}_{xx}(t-1)\,\boldsymbol{x}(t)]$.

Let $\underline{\boldsymbol{W}}(t)$ be a $n \times (r+p)$ orthonormal matrix spanning the range space of $\underline{\boldsymbol{V}}(t)$. Then $\boldsymbol{W}(t)$ will be written in the form

$$\boldsymbol{W}(t) = \underline{\boldsymbol{W}}(t)\,\boldsymbol{U}(t). \tag{3}$$

where $\boldsymbol{U}(t)$ is a $(r+p)\times r$ orthonormal matrix. In this case

$$\mathcal{J}\left(\boldsymbol{W}(t)\right) = \operatorname{trace}\left(\boldsymbol{U}(t)^H\underline{\boldsymbol{C}}_{yy}(t)\boldsymbol{U}(t)\right) \tag{4}$$

where $\underline{\boldsymbol{C}}_{yy}(t)$ is the $(r+p)\times(r+p)$ matrix

$$\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{W}}(t)^H\,\boldsymbol{C}_{xx}(t)\,\underline{\boldsymbol{W}}(t). \tag{5}$$

Thus the exhaustive search among all $n\times r$ orthonormal matrices $\boldsymbol{W}(t)$ is replaced by the maximization of (4) over all $(r+p)\times p$ orthonormal matrices $\boldsymbol{U}(t)$. The result of this maximization is well-known: $\boldsymbol{U}(t)$ must span the r-dimensional major subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Thus the subspace weighting matrix $\boldsymbol{W}(t)$ can be tracked by computing

- an orthonormal basis $\underline{\boldsymbol{W}}(t)$ of the range space of $\underline{\boldsymbol{V}}(t)$,
- the matrix $\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{W}}(t)^H\,\boldsymbol{C}_{xx}(t)\,\underline{\boldsymbol{W}}(t)$,
- a $(r+p)\times r$ orthonormal matrix $\boldsymbol{U}(t)$ spanning the r-dimensional major subspace of $\underline{\boldsymbol{C}}_{yy}(t)$
- the matrix $\boldsymbol{W}(t) = \underline{\boldsymbol{W}}(t)\,\boldsymbol{U}(t)$.

In particular, $\boldsymbol{U}(t)$ can be obtained via the eigenvalue decomposition of $\underline{\boldsymbol{C}}_{yy}(t)$. As a consequence, the columns of the resulting matrix $\boldsymbol{W}(t)$ defined in equation (3) correspond to the r major eigenvectors of $\boldsymbol{C}_{xx}(t)$. However, this calculation would lead to an overall complexity of $O(nr^2)$, like in [1].

In order to reduce the global complexity to $O(nr)$, we choose a different strategy which avoids the eigenvalue decomposition. As mentioned above, $\boldsymbol{U}(t)$ must be an orthonormal matrix spanning the r-dimensional major subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Therefore $\boldsymbol{U}(t)$ can be obtained as an orthogonal complement of the p-dimensional major subspace of $\underline{\boldsymbol{Z}}(t) = \underline{\boldsymbol{C}}_{yy}(t)^{-1}$. Thus the YAST algorithm computes $\underline{\boldsymbol{Z}}(t)$ and its p-dimensional major subspace, and computes $\boldsymbol{U}(t)$ as an orthogonal complement of this subspace.

As shown in section 3, this algorithm can be efficiently implemented by updating the inverse $\boldsymbol{Z}(t)$ of the $r \times r$ compressed correlation matrix $\boldsymbol{C}_{yy}(t)$, defined as

$$\boldsymbol{C}_{yy}(t) = \boldsymbol{W}(t)^H \boldsymbol{C}_{xx}(t) \boldsymbol{W}(t). \tag{6}$$

3. FAST IMPLEMENTATION OF YAST

Below, a fast implementation of YAST is proposed, whose global cost is only $(3p+1)nr$ flops[1]. It can be decomposed into four steps: computation of $\underline{\boldsymbol{W}}(t)$ (section 3.1), computation of $\underline{\boldsymbol{Z}}(t)$ (section 3.2), update of $\boldsymbol{W}(t)$ (section 3.3), update of $\boldsymbol{Z}(t)$ (section 3.4). This implementation is summarized in section 3.5.

3.1. Computation of $\underline{\boldsymbol{W}}(t)$

Define the $r \times p$ matrix $\underline{\boldsymbol{y}}(t) = \boldsymbol{W}(t-1)^H \underline{\boldsymbol{v}}(t)$ and let

$$\underline{\boldsymbol{e}}(t) = \underline{\boldsymbol{v}}(t) - \boldsymbol{W}(t-1)\, \underline{\boldsymbol{y}}(t). \tag{7}$$

The $n \times p$ matrix $\underline{\boldsymbol{e}}(t)$ is orthogonal to the range space of $\boldsymbol{W}(t-1)$. Let $\underline{\sigma}(t)$ be a square root of the $p \times p$ matrix $\underline{\boldsymbol{e}}(t)^H \underline{\boldsymbol{e}}(t)$:

$$\underline{\sigma}(t) = \left(\underline{\boldsymbol{e}}(t)^H \underline{\boldsymbol{e}}(t)\right)^{\frac{1}{2}} = \left(\underline{\boldsymbol{v}}(t)^H \underline{\boldsymbol{v}}(t) - \underline{\boldsymbol{y}}(t)^H \underline{\boldsymbol{y}}(t)\right)^{\frac{1}{2}}. \tag{8}$$

Below, we suppose that $\underline{\sigma}(t)$ is non-singular. Then the matrix

$$\underline{\boldsymbol{W}}(t) = \left[\boldsymbol{W}(t-1),\, \underline{\boldsymbol{e}}(t)\, \underline{\sigma}(t)^{-1}\right] \tag{9}$$

is orthonormal. In particular, $\underline{\boldsymbol{V}}(t)$ can be written in the form

$$\underline{\boldsymbol{V}}(t) = \underline{\boldsymbol{W}}(t)\, \underline{\boldsymbol{R}}(t) \tag{10}$$

where $\underline{\boldsymbol{R}}(t)$ is the $(r+p) \times (r+p)$ non-singular matrix

$$\underline{\boldsymbol{R}}(t) = \left[\begin{array}{c|c} \boldsymbol{I}_r & \underline{\boldsymbol{y}}(t) \\ \hline \boldsymbol{0}_{p \times r} & \underline{\sigma}(t) \end{array}\right]. \tag{11}$$

3.2. Computation of $\underline{\boldsymbol{Z}}(t)$

As mentioned in section 2, the matrix $\underline{\boldsymbol{Z}}(t)$ is defined as the inverse of the $(r+p) \times (r+p)$ matrix $\underline{\boldsymbol{C}}_{yy}(t)$ defined in equation (5). Therefore, computing $\underline{\boldsymbol{Z}}(t)$ consists in computing $\underline{\boldsymbol{C}}_{yy}(t)$ (section 3.2.1), before inverting $\underline{\boldsymbol{C}}_{yy}(t)$ (section 3.2.2).

3.2.1. Computation of $\underline{\boldsymbol{C}}_{yy}(t)$

Substituting equation (10) into equation (5) yields

$$\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{R}}(t)^{-H} \underline{\boldsymbol{C}}'_{yy}(t)\, \underline{\boldsymbol{R}}(t)^{-1} \tag{12}$$

where $\underline{\boldsymbol{C}}'_{yy}(t)$ is the $(r+p) \times (r+p)$ matrix

$$\underline{\boldsymbol{C}}'_{yy}(t) = \underline{\boldsymbol{V}}(t)^H \boldsymbol{C}_{xx}(t)\, \underline{\boldsymbol{V}}(t). \tag{13}$$

[1]In this paper, a flop is a multiply / accumulate (MAC) operation.

Then substituting equations (1) and (2) into equation (13) yields

$$\underline{\boldsymbol{C}}'_{yy}(t) = \left[\begin{array}{c|c} \widetilde{\boldsymbol{C}}_{yy}(t) & \underline{\boldsymbol{y}}''(t) \\ \hline \underline{\boldsymbol{y}}''(t)^H & \underline{c}_{yy}(t) \end{array}\right] \tag{14}$$

where

$$\begin{aligned}
\widetilde{\boldsymbol{C}}_{yy}(t) &= \beta\, \boldsymbol{C}_{yy}(t-1) + \boldsymbol{y}(t)\, \boldsymbol{y}(t)^H && (15)\\
\underline{\boldsymbol{y}}''(t) &= \beta\, \underline{\boldsymbol{y}}'(t) + \boldsymbol{y}(t)\, \underline{\boldsymbol{\alpha}}(t) && (16)\\
\underline{c}_{yy}(t) &= \beta\, \underline{\boldsymbol{v}}(t)^H \underline{\boldsymbol{v}}'(t) + \underline{\boldsymbol{\alpha}}(t)^H \underline{\boldsymbol{\alpha}}(t) && (17)\\
\boldsymbol{y}(t) &= \boldsymbol{W}(t-1)^H \boldsymbol{x}(t) && (18)\\
\underline{\boldsymbol{y}}'(t) &= \boldsymbol{W}(t-1)^H \underline{\boldsymbol{v}}'(t) && (19)\\
\underline{\boldsymbol{\alpha}}(t) &= \boldsymbol{x}(t)^H \underline{\boldsymbol{v}}(t) && (20)\\
\underline{\boldsymbol{v}}'(t) &= \boldsymbol{C}_{xx}(t-1)\, \underline{\boldsymbol{v}}(t). && (21)
\end{aligned}$$

3.2.2. Inversion of $\underline{\boldsymbol{C}}_{yy}(t)$

The matrix $\underline{\boldsymbol{C}}_{yy}(t)$ can be obtained from $\underline{\boldsymbol{C}}'_{yy}(t)$ by means of equation (12). Thus inverting $\underline{\boldsymbol{C}}_{yy}(t)$ requires inverting $\underline{\boldsymbol{C}}'_{yy}(t)$. As shown below, this last operation can be achieved by first inverting the $r \times r$ upper left corner of $\underline{\boldsymbol{C}}'_{yy}(t)$, denoted $\widetilde{\boldsymbol{C}}_{yy}(t)$.

3.2.2.1. Inversion of $\widetilde{\boldsymbol{C}}_{yy}(t)$

Suppose that $\boldsymbol{C}_{yy}(t-1)$ is non-singular and let $\boldsymbol{Z}(t-1) = \boldsymbol{C}_{yy}(t-1)^{-1}$. Let $\boldsymbol{h}(t) = \boldsymbol{Z}(t-1)\, \boldsymbol{y}(t)$. By applying lemma 1 to equation (15), it can be shown that $\widetilde{\boldsymbol{C}}_{yy}(t)$ is non-singular if and only if $\beta + \boldsymbol{y}(t)^H \boldsymbol{h}(t)$ is non-singular[2]. In this case, let $\gamma(t) = \left(\beta + \boldsymbol{y}(t)^H \boldsymbol{h}(t)\right)^{-1}$. Then $\widetilde{\boldsymbol{Z}}(t) \triangleq \widetilde{\boldsymbol{C}}_{yy}(t)^{-1}$ satisfies

$$\widetilde{\boldsymbol{Z}}(t) = \frac{1}{\beta}\left(\boldsymbol{Z}(t-1) - \boldsymbol{h}(t)\, \gamma(t)\, \boldsymbol{h}(t)^H\right). \tag{22}$$

3.2.2.2. Inversion of $\underline{\boldsymbol{C}}_{yy}(t)$

Inversing equation (12) yields

$$\underline{\boldsymbol{Z}}(t) = \underline{\boldsymbol{R}}(t)\, \underline{\boldsymbol{C}}'_{yy}(t)^{-1} \underline{\boldsymbol{R}}(t)^H \tag{23}$$

Let $\underline{\boldsymbol{h}}(t) = \widetilde{\boldsymbol{Z}}(t)\, \underline{\boldsymbol{y}}''(t)$. By applying lemma 2 to equation (14), it can be shown that $\underline{\boldsymbol{C}}'_{yy}(t)$ is non-singular if and only if $\underline{c}_{yy}(t) - \underline{\boldsymbol{y}}''(t)^H \underline{\boldsymbol{h}}(t)$ is non-singular[3]. In this case,

$$\underline{\boldsymbol{C}}'_{yy}(t)^{-1} = \left[\begin{array}{c|c} \widetilde{\boldsymbol{Z}}(t) + \underline{\boldsymbol{h}}(t)\, \underline{\gamma}(t) \underline{\boldsymbol{h}}(t)^H & -\underline{\boldsymbol{h}}(t) \underline{\gamma}(t) \\ \hline -\underline{\gamma}(t) \underline{\boldsymbol{h}}(t)^H & \underline{\gamma}(t) \end{array}\right] \tag{24}$$

where $\underline{\gamma}(t) = \left(\underline{c}_{yy}(t) - \underline{\boldsymbol{y}}''(t)^H \underline{\boldsymbol{h}}(t)\right)^{-1}$.

Substituting equations (11) and (24) into equation (23) yields

$$\underline{\boldsymbol{Z}}(t) = \left[\begin{array}{c|c} \widetilde{\boldsymbol{Z}}'(t) & -\underline{\boldsymbol{g}}(t) \\ \hline -\underline{\boldsymbol{g}}(t)^H & \underline{\gamma}'(t) \end{array}\right] \tag{25}$$

[2]Lemma 1 can be found in the appendix. It must be applied with $\boldsymbol{C} = \boldsymbol{C}_{yy}(t-1)$, $\boldsymbol{A} = \boldsymbol{y}(t)$, $\boldsymbol{B} = \boldsymbol{y}(t)^H$, and $\boldsymbol{D} = 1$.

[3]Lemma 2 can be found in the appendix. It must be applied with $\boldsymbol{C} = \widetilde{\boldsymbol{C}}_{yy}(t)$, $\boldsymbol{A} = \underline{\boldsymbol{y}}''(t)$, $\boldsymbol{B} = \underline{\boldsymbol{y}}''(t)^H$, and $\boldsymbol{D} = \underline{c}_{yy}(t)$.

where

$$\underline{\boldsymbol{h}}'(t) = \underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{y}}(t) \tag{26}$$

$$\widetilde{\boldsymbol{Z}}'(t) = \widetilde{\boldsymbol{Z}}(t) + \underline{\boldsymbol{h}}'(t)\,\underline{\gamma}(t)\,\underline{\boldsymbol{h}}'(t)^H \tag{27}$$

$$\underline{\boldsymbol{g}}(t) = \underline{\boldsymbol{h}}'(t)\,\underline{\gamma}(t)\,\underline{\sigma}(t)^H \tag{28}$$

$$\underline{\gamma}'(t) = \underline{\sigma}(t)\,\underline{\gamma}(t)\,\underline{\sigma}(t)^H. \tag{29}$$

3.3. Update of $\boldsymbol{W}(t)$

Let $\underline{\boldsymbol{\phi}}(t)$ be a $(r+p)\times p$ orthonormal matrix whose columns span the p-dimensional major subspace of the positive definite matrix $\underline{\boldsymbol{Z}}(t)$ of dimension $(r+p)\times(r+p)$. In particular, there is a $p\times p$ positive definite matrix $\underline{\lambda}(t)$ such that

$$\underline{\boldsymbol{Z}}(t)\,\underline{\boldsymbol{\phi}}(t) = \underline{\boldsymbol{\phi}}(t)\,\underline{\lambda}(t). \tag{30}$$

Let $\underline{\boldsymbol{\varphi}}(t)$ be the $r\times p$ matrix containing the r first rows of $\underline{\boldsymbol{\phi}}(t)$, and $\underline{z}(t)$ be the $p\times p$ matrix containing its p last rows:

$$\underline{\boldsymbol{\phi}}(t)^T = \left[\underline{\boldsymbol{\varphi}}(t)^T,\, \underline{z}(t)^T\right]. \tag{31}$$

Let $\underline{z}(t) = \underline{\rho}(t)\,\underline{\theta}(t)$ be the polar decomposition of $\underline{z}(t)$, where $\underline{\rho}(t)$ is positive definite and $\underline{\theta}(t)$ is orthonormal. Let[4]

$$\underline{\boldsymbol{f}}(t) = \underline{\boldsymbol{\varphi}}(t)\,\underline{\theta}(t)^H \tag{32}$$

$$\underline{\boldsymbol{f}}'(t) = \underline{\boldsymbol{f}}(t)\left(\underline{I} + \underline{\rho}(t)\right)^{-1}. \tag{33}$$

Then it can be readily verified that the $(r+p)\times r$ matrix

$$\boldsymbol{U}(t) = \left[\begin{array}{c} \boldsymbol{I}_r - \underline{\boldsymbol{f}}'(t)\,\underline{\boldsymbol{f}}(t)^H \\ \hline -\underline{\boldsymbol{f}}(t)^H \end{array}\right] \tag{34}$$

is orthonormal and satisfies $\boldsymbol{U}(t)^H\underline{\boldsymbol{\phi}}(t) = \boldsymbol{0}$. Therefore $\boldsymbol{U}(t)$ is an orthonormal basis of the r-dimensional minor subspace of $\underline{\boldsymbol{Z}}(t)$.

Substituting equations (9) and (34) into equation (3) shows a recursion for the subspace weighting matrix:

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) - \underline{\boldsymbol{e}}'(t)\,\underline{\boldsymbol{f}}(t)^H \tag{35}$$

where $\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{e}}(t)\,\underline{\sigma}(t)^{-1} + \boldsymbol{W}(t-1)\underline{\boldsymbol{f}}'(t)$.

Finally, substituting equation (7) into this last definition yields

$$\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{v}}(t)\,\underline{\sigma}(t)^{-1} - \boldsymbol{W}(t-1)\underline{\boldsymbol{y}}'''(t) \tag{36}$$

where $\underline{\boldsymbol{y}}'''(t) = \underline{\boldsymbol{y}}(t)\,\underline{\sigma}(t)^{-1} - \underline{\boldsymbol{f}}'(t)$.

3.4. Update of $\boldsymbol{Z}(t)$

The auxiliary matrix $\boldsymbol{Z}(t)$ can also be efficiently updated. Indeed, substituting equations (3) and (5) into equation (6) yields $\boldsymbol{C}_{yy}(t) = \boldsymbol{U}(t)^H\underline{\boldsymbol{C}}_{yy}(t)\boldsymbol{U}(t)$. As the orthonormal matrix $\boldsymbol{U}(t)$ spans an invariant subspace of $\underline{\boldsymbol{C}}_{yy}(t)$, this last equation yields

$$\boldsymbol{Z}(t) = \boldsymbol{U}(t)^H\underline{\boldsymbol{Z}}(t)\boldsymbol{U}(t). \tag{37}$$

Substituting equations (33), (32), and (31) into (34) shows that

$$\boldsymbol{U}(t) = \left[\begin{array}{c} \boldsymbol{I}_r \\ \hline -\underline{\boldsymbol{f}}'(t)^H \end{array}\right] - \underline{\boldsymbol{\phi}}(t)\,\underline{\theta}(t)^H\,\underline{\boldsymbol{f}}'(t)^H. \tag{38}$$

Thus substituting equations (38), (25), (30) into (37) yields

$$\boldsymbol{Z}(t) = \widetilde{\boldsymbol{Z}}'(t) + \underline{\boldsymbol{g}}'(t)\,\underline{\boldsymbol{f}}'(t)^H + \underline{\boldsymbol{f}}'(t)\,\underline{\boldsymbol{g}}(t)^H \tag{39}$$

where $\underline{\boldsymbol{g}}'(t) = \underline{\boldsymbol{g}}(t) + \underline{\boldsymbol{f}}'(t)\left(\underline{\gamma}'(t) - \underline{\theta}(t)\underline{\lambda}(t)\underline{\theta}(t)^H\right)$.

Table 1. Pseudo-code of the YAST algorithm

	eq.:	**flops:**
$\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}(t)$		nr
$\boldsymbol{x}'(t) = \boldsymbol{C}_{xx}(t-1)\,\boldsymbol{x}(t)$		$9n$
$\boldsymbol{y}'(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}'(t)$		nr
switch p {		
case 1: $\underline{\boldsymbol{v}}(t) = \boldsymbol{x}(t)$, $\underline{\boldsymbol{v}}'(t) = \boldsymbol{x}'(t)$,		
$\underline{\boldsymbol{y}}(t) = \boldsymbol{y}(t)$, $\underline{\boldsymbol{y}}'(t) = \boldsymbol{y}'(t)$.		
case 2: $\boldsymbol{x}''(t) = \boldsymbol{C}_{xx}(t-1)^2\,\boldsymbol{x}(t)$,		$32n$
$\boldsymbol{y}''(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}''(t)$,		nr
$\underline{\boldsymbol{v}}(t) = \left[\boldsymbol{x}(t), \boldsymbol{x}'(t)\right]$, $\underline{\boldsymbol{v}}'(t) = \left[\boldsymbol{x}'(t), \boldsymbol{x}''(t)\right]$,		
$\underline{\boldsymbol{y}}(t) = \left[\boldsymbol{y}(t), \boldsymbol{y}'(t)\right]$, $\underline{\boldsymbol{y}}'(t) = \left[\boldsymbol{y}'(t), \boldsymbol{y}''(t)\right]$.}		
$\underline{\sigma}(t) = \left(\underline{\boldsymbol{v}}(t)^H\underline{\boldsymbol{v}}(t) - \underline{\boldsymbol{y}}(t)^H\underline{\boldsymbol{y}}(t)\right)^{\frac{1}{2}}$	(8)	p^2n
$\boldsymbol{h}(t) = \boldsymbol{Z}(t-1)\,\boldsymbol{y}(t)$		r^2
$\gamma(t) = \left(\beta + \boldsymbol{y}(t)^H\boldsymbol{h}(t)\right)^{-1}$		r
$\widetilde{\boldsymbol{Z}}(t) = \frac{1}{\beta}\left(\boldsymbol{Z}(t-1) - \boldsymbol{h}(t)\,\gamma(t)\,\boldsymbol{h}(t)^H\right)$	(22)	r^2
$\underline{\boldsymbol{\alpha}}(t) = \boldsymbol{x}(t)^H\underline{\boldsymbol{v}}(t)$	(20)	pn
$\underline{\boldsymbol{y}}''(t) = \beta\underline{\boldsymbol{y}}'(t) + \boldsymbol{y}(t)\,\underline{\boldsymbol{\alpha}}(t)$	(16)	pr
$\underline{c}_{yy}(t) = \beta\underline{\boldsymbol{v}}(t)^H\underline{\boldsymbol{v}}'(t) + \underline{\boldsymbol{\alpha}}(t)^H\underline{\boldsymbol{\alpha}}(t)$	(17)	p^2n
$\underline{\boldsymbol{h}}(t) = \widetilde{\boldsymbol{Z}}(t)\,\underline{\boldsymbol{y}}''(t)$		pr^2
$\underline{\gamma}(t) = \left(\underline{c}_{yy}(t) - \underline{\boldsymbol{y}}''(t)^H\underline{\boldsymbol{h}}(t)\right)^{-1}$		p^2r
$\underline{\boldsymbol{h}}'(t) = \underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{y}}(t)$	(26)	
$\widetilde{\boldsymbol{Z}}'(t) = \widetilde{\boldsymbol{Z}}(t) + \underline{\boldsymbol{h}}'(t)\,\underline{\gamma}(t)\,\underline{\boldsymbol{h}}'(t)^H$	(27)	pr^2
$\underline{\boldsymbol{g}}(t) = \underline{\boldsymbol{h}}'(t)\,\underline{\gamma}(t)\,\underline{\sigma}(t)^H$	(28)	p^2r
$\underline{\gamma}'(t) = \underline{\sigma}(t)\,\underline{\gamma}(t)\,\underline{\sigma}(t)^H$	(29)	p^3
$\underline{\boldsymbol{Z}}(t) = \left[\widetilde{\boldsymbol{Z}}'(t), -\underline{\boldsymbol{g}}(t); -\underline{\boldsymbol{g}}(t)^H, \underline{\gamma}'(t)\right]$	(25)	
$\left(\underline{\boldsymbol{\phi}}(t), \underline{\lambda}(t)\right) = \mathrm{eigs}\left(\underline{\boldsymbol{Z}}(t), p\right)$	(30)	$O(pr^2)$
$\left[\underline{\boldsymbol{\varphi}}(t)^T, \underline{z}(t)^T\right] = \underline{\boldsymbol{\phi}}(t)^T$	(31)	
$\left(\underline{\rho}(t), \underline{\theta}(t)\right) = \mathrm{polar}\left(\underline{z}(t)\right)$		$O(p^3)$
$\underline{\boldsymbol{f}}(t) = \underline{\boldsymbol{\varphi}}(t)\,\underline{\theta}(t)^H$	(32)	p^2r
$\underline{\boldsymbol{f}}'(t) = \underline{\boldsymbol{f}}(t)\left(\underline{I} + \underline{\rho}(t)\right)^{-1}$	(33)	p^2r
$\underline{\boldsymbol{y}}'''(t) = \underline{\boldsymbol{y}}(t)\,\underline{\sigma}(t)^{-1} - \underline{\boldsymbol{f}}'(t)$		p^2r
$\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{v}}(t)\underline{\sigma}(t)^{-1} - \boldsymbol{W}(t-1)\underline{\boldsymbol{y}}'''(t)$	(36)	pnr
$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) - \underline{\boldsymbol{e}}'(t)\,\underline{\boldsymbol{f}}(t)^H$	(35)	pnr
$\underline{\boldsymbol{g}}'(t) = \underline{\boldsymbol{g}}(t) + \underline{\boldsymbol{f}}'(t)\left(\underline{\gamma}'(t) - \underline{\theta}(t)\underline{\lambda}(t)\underline{\theta}(t)^H\right)$		p^2r
$\boldsymbol{Z}(t) = \widetilde{\boldsymbol{Z}}'(t) + \underline{\boldsymbol{g}}'(t)\,\underline{\boldsymbol{f}}'(t)^H + \underline{\boldsymbol{f}}'(t)\,\underline{\boldsymbol{g}}(t)^H$	(39)	$2pr^2$
$\boldsymbol{Z}(t) = \frac{\boldsymbol{Z}(t)+\boldsymbol{Z}(t)^H}{2}$		r^2
Total:		$(3p+1)nr + O(n+r^2)$

3.5. Implementation

The complete pseudo-code of YAST is presented in table 1[5]. Its global cost is $4nr$ flops in the case $p=1$, which is lower or equal to that of FST, NIC and OPAST , and $7nr$ in the case $p=2$, which remains one order of magnitude lower than that of Karasalo's algorithm, Loraf and SP. The computation of the vectors $\boldsymbol{x}'(t) = \boldsymbol{C}_{xx}(t-1)\,\boldsymbol{x}(t)$ and $\boldsymbol{x}''(t) = \boldsymbol{C}_{xx}(t-1)^2\boldsymbol{x}(t)$ is reduced from $O(n^2)$ to $O(n)$ by means of the technique described in [1], which exploits the shift invariance property of the correlation matrix.

[4]Since $\underline{\rho}(t)$ is positive definite, the $p\times p$ matrix $\underline{I}+\underline{\rho}(t)$ is also positive definite. In particular, $\underline{I}+\underline{\rho}(t)$ is non-singular.

[5]We observed that the YAST algorithm is prone to numerical instability if implemented as proposed above. This is due to a loss of symmetry of the matrix $\boldsymbol{Z}(t)$. To make YAST stable, the symmetry must be enforced at the end of each iteration (see table 1).

4. SIMULATION RESULTS

In this section, the performance of the subspace estimation is analyzed in terms of the maximum principal angle between the true major subspace of the correlation matrix $\boldsymbol{C}_{xx}(t)$ (obtained via an exact eigenvalue decomposition), and the estimated major subspace of the same correlation matrix (obtained with the subspace tracker). This error criterion was initially proposed by P. Comon and G.H. Golub as a measure of the distance between equidimensional subspaces [9]. The test signal is a sum of $r = 4$ complex sinusoidal sources plus a complex white gaussian noise (the SNR is 5.7 dB). The frequencies of the sinusoids vary according to a jump scenario originally proposed by P. Strobach [10]: their values abruptly change at different time instants, between which they remain constant. Their variations are represented on Figure 1-a.

Figure 1-b shows the maximum principal angle error trajectory obtained with the orthonormal version of PAST, referred to as OPAST [8], with parameters $n = 80$ and $\beta \approx 0.99$. We chose the OPAST subspace tracker for performance comparison, because we observed in [2] that none of the other algorithms mentioned in the introduction outperformed OPAST in terms of subspace estimation. This result is to be compared to that obtained with the YAST algorithm with the same parameters (figure 1-c) and $p = 1$. It can be noticed that YAST converges much faster than OPAST. Finally, figure 1-d shows the result obtained with YAST in the case $p = 2$. The convergence is even faster than in the previous case.

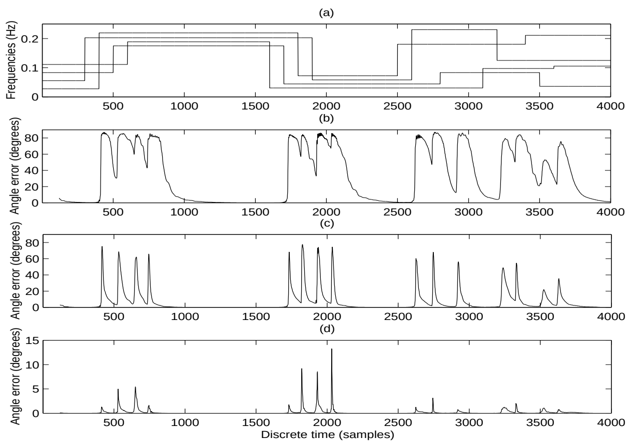

Fig. 1. Simulation results

5. CONCLUSIONS

In this paper, a new algorithm for subspace tracking was presented, which is derived from the SP algorithm by C.E. Davila. In particular, the proof for its convergence can be found in [1]. This algorithm reaches the linear complexity $O(nr)$ and greatly outperforms classical subspace trackers of the same complexity. Moreover, it guarantees the orthonormality of the subspace weighting matrix at each time step. Note that it can be transformed to track the minor subspace of the correlation matrix[6]. It can also be transformed to track the major (or minor) subspace of a correlation matrix updated on a truncated window of finite length[7].

[6]In this case, the matrix $\boldsymbol{U}(t)$ must span the r-dimensional minor subspace of $\underline{\boldsymbol{C}}_{yy}(t)$ (instead of $\underline{\boldsymbol{Z}}(t)$).

[7]In this case, the column vector $\boldsymbol{x}(t - l)$ must be appended to the directional search matrix $\underline{v}(t)$, where l is the window length.

6. APPENDIX

The following lemma shows how the inverse of a matrix changes upon a small-rank adjustment [11, pp. 18-19].

Lemma 1 (Inverse of a small-rank adjustment). *Let $r \in \mathbb{N}$ and $\boldsymbol{C}$ be a $r \times r$ non-singular matrix. Let $q \in \mathbb{N}$, $\boldsymbol{A}$ be a $r \times q$ matrix, $\boldsymbol{B}$ be a $q \times r$ matrix, and $\boldsymbol{D}$ be a $q \times q$ non-singular matrix. Define the $r \times r$ matrix $\widetilde{\boldsymbol{C}} = \boldsymbol{C} + \boldsymbol{A}\,\boldsymbol{D}\,\boldsymbol{B}$. Then $\widetilde{\boldsymbol{C}}$ is non-singular if and only if the $q \times q$ matrix $\boldsymbol{D}^{-1} + \boldsymbol{B}\,\boldsymbol{C}^{-1}\boldsymbol{A}$ is non-singular, and in this case $\widetilde{\boldsymbol{C}}^{-1} = \boldsymbol{C}^{-1} - \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{B}\boldsymbol{C}^{-1}$, where $\boldsymbol{\Gamma} = \left(\boldsymbol{D}^{-1} + \boldsymbol{B}\,\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1}$.*

Similarly, the following lemma shows how the inverse of a matrix changes upon a low-dimensional expansion (the proof is omitted to not exceed the authorized number of pages).

Lemma 2 (Inverse of a low-dimensional expansion). *Let $r \in \mathbb{N}$ and $\boldsymbol{C}$ be a $r \times r$ non-singular matrix. Let $p \in \mathbb{N}$, $\boldsymbol{A}$ be a $r \times p$ matrix, $\boldsymbol{B}$ be a $p \times r$ matrix, and $\boldsymbol{D}$ be a $p \times p$ matrix. Define the $(r+p) \times (r+p)$ matrix $\underline{\boldsymbol{C}} = \left[\begin{array}{c|c} \boldsymbol{C} & \boldsymbol{A} \\ \hline \boldsymbol{B} & \boldsymbol{D} \end{array}\right]$. Then $\underline{\boldsymbol{C}}$ is non-singular if and only if the $p \times p$ matrix $\boldsymbol{D} - \boldsymbol{B}\,\boldsymbol{C}^{-1}\boldsymbol{A}$ is non-singular. In this case let $\boldsymbol{\Gamma} = \left(\boldsymbol{D} - \boldsymbol{B}\,\boldsymbol{C}^{-1}\boldsymbol{A}\right)^{-1}$. Then*

$$\underline{\boldsymbol{C}}^{-1} = \left[\begin{array}{c|c} \boldsymbol{C}^{-1} + \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{B}\boldsymbol{C}^{-1} & -\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Gamma} \\ \hline -\boldsymbol{\Gamma}\boldsymbol{B}\boldsymbol{C}^{-1} & \boldsymbol{\Gamma} \end{array}\right]. \qquad (40)$$

7. REFERENCES

[1] C. E. Davila, “Efficient, high performance, subspace tracking for time-domain data,” *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp. 3307–3315, Dec. 2000.

[2] R. Badeau, B. David, and G. Richard, “Fast Approximated Power Iteration Subspace Tracking,” *IEEE Trans. Signal Processing*, to be published.

[3] I. Karasalo, “Estimating the covariance matrix by signal subspace averaging,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 8–12, Feb. 1986.

[4] B. Yang, “Projection Approximation Subspace Tracking,” *IEEE Trans. Signal Processing*, vol. 44, no. 1, Jan. 1995.

[5] P. Strobach, “Low-rank adaptive filters,” *IEEE Trans. Signal Processing*, vol. 44, no. 12, pp. 2932–2947, Dec. 1996.

[6] D. J. Rabideau, “Fast, rank adaptive subspace tracking and applications,” *IEEE Trans. Signal Processing*, vol. 44, no. 9, pp. 2229–2244, Sept. 1996.

[7] Y. Miao and Y. Hua, “Fast subspace tracking and neural network learning by a novel information criterion,” *IEEE Trans. Signal Processing*, vol. 46, no. 7, pp. 1967–1979, July 1998.

[8] K. Abed-Meraim, A. Chkeif, and Y. Hua, “Fast orthonormal PAST algorithm,” *IEEE Signal Proc. Letters*, vol. 7, no. 3, pp. 60–62, Mar. 2000.

[9] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, third edition, 1996.

[10] P. Strobach, “Fast recursive subspace adaptive ESPRIT algorithms,” *IEEE Trans. Signal Processing*, vol. 46, no. 9, pp. 2413–2430, Sept. 1998.

[11] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1985.