AN EFFICIENT & STABLE ALGORITHM FOR MINOR SUBSPACE TRACKING AND STABILITY ANALYSIS

S. Bartelmaos & K. Abed-Meraim

ENST-Paris, TSI Department , 37/39 rue Dareau 75014, Paris Cedex 14, France

ABSTRACT

In this paper, we present a theoretical stability analysis of the YAST algorithm used for tracking the noise subspace of the covariance matrix associated with time series. This analysis demonstrates the instability of the YAST and a more stable alternative solution is proposed. In addition to its stability, the resulting algorithm is less expensive than the YAST and has a computational complexity of order $O(nr)$ flops per iteration where n is the size of the observation vector and $r < n$ is the minor subspace dimension. Finally, we pay a special attention to the case $r = 1$ due to its importance in many quadratic optimization problems. In that particular case, we propose a simplified version of the algorithm to estimate either the first principal eigenvector or the last minor eigenvector of the covariance matrix. Simulation results are provided at the end to validate the theoretical stability analysis and to illustrate the tracking capacity of the proposed algorithm.

Index Terms— Adaptive estimation, fast algorithm, numerical stability.

1. INTRODUCTION

Principal and minor subspace analysis (PSA and MSA) are two important subspace-based high resolution methods that have been applied successfully to both temporal and spatial domain spectral analysis such as: adaptive filtering, multiuser detection in CDMA [1], mobile positioning [2] and blind channel equalization [3].

Many subspace tracking algorithms exist in the literature that can be classified according to their computational complexity. Since usually $r << n$, schemes requiring $O(n^2)$ or $O(n^2 r)$ operations will be classified as high complexity; algorithms with complexity $O(nr^2)$ as medium complexity, finally algorithmic schemes requiring only $O(nr)$ operations are said of low complexity. Recently, a new low complexity subspace tracker, referred to as the YAST (Yet Another Subspace Tracking algorithm) has been introduced in [4, 5] as a generalization of Davila's algorithm in [6] for PSA and MSA. This algorithm greatly outperforms many well known subspace trackers in terms of subspace estimation, such as Karasalos algorithm [7], PAST [8], Loraf [9] and OPAST [10].

In this paper, we propose first a simplified version of YAST which extracts the first principal (resp. minor) component of a covariance matrix. Indeed, this particular case of subspace tracking is of high importance in many practical problems where quadratic optimization is used. The resulting algorithm is very easy to implement and is referred to as the MYAST (Modified YAST) algorithm.

Secondly, we focus on the algorithm's numerical stability when used for the minor subspace extraction. We derived a theoretical analysis that proves the numerical instability of the algorithm. Then, we propose a fast orthogonalization method based on the Pairwise Gram Schmidt (PGS) technique [11] in order to stabilize the algorithm. The proposed algorithm referred to as YAST-PGS is slightly less expensive than the YAST. Simulation results highlight the relative good stability behavior of YAST-PGS.

2. MYAST

Let $\mathbf{x}(t)$ be a sequence of $n \times 1$ random vectors with covariance matrix $\mathbf{C} = E[\mathbf{x}(t)\mathbf{x}(t)^H]$. This matrix can be recursively updated according to

$$\mathbf{C}(t) = \beta\mathbf{C}(t-1) + \mathbf{x}(t)\mathbf{x}(t)^H \tag{1}$$

where $0 < \beta < 1$ is a forgetting factor. In [4] (resp. [5]), an efficient algorithm (the YAST) has been proposed for the extraction of the subspace spanned by the $r < n$ principal (resp. minor) eigenvectors of the covariance matrix. We consider here the particular case $r = 1$ and propose a modified version of YAST to estimate the first principal (minor) eigenvector of the covariance matrix. As shown in [4, 5], the principal (minor) eigenvector $\mathbf{w}(t)$ is obtained by maximization (minimization) of the criterion

$$J(\mathbf{w}(t)) = \mathbf{w}(t)^H\mathbf{C}(t)\mathbf{w}(t). \tag{2}$$

However, a brute force implementation of this optimization problem is computationally demanding (the complexity is of $O(n^2)$), and does not lead to a simple recursion between $\mathbf{w}(t)$ and $\mathbf{w}(t-1)$. In order to reduce the computational cost, the idea introduced in [6] consists in limiting this search into the space spanned by vector $\mathbf{w}(t-1)$ plus one additional search direction given by the current observation vector. In other words, the principal eigenvector $\mathbf{w}(t)$ is to be found as a subspace vector of the 2-dimensional space spanned by the $n \times 2$ matrix

$$\mathbf{V}(t) = [\mathbf{w}(t-1), \mathbf{x}(t)]\,. \tag{3}$$

Let $\underline{\mathbf{V}}(t)$ be a $n \times 2$ orthonormal matrix spanning the range space of $\mathbf{V}(t)$. Then $\mathbf{w}(t)$ will be written in the form

$$\mathbf{w}(t) = \underline{\mathbf{V}}(t)\mathbf{u}(t) \tag{4}$$

where $\mathbf{u}(t)$ is a 2-elements unitary vector. In this case

$$J(\mathbf{w}(t)) = \mathbf{u}(t)^H\underline{\mathbf{C}}(t)\mathbf{u}(t) \tag{5}$$

where $\underline{\mathbf{C}}(t)$ is the 2×2 matrix

$$\underline{\mathbf{C}}(t) = \underline{\mathbf{V}}(t)^H\mathbf{C}(t)\underline{\mathbf{V}}(t). \tag{6}$$

The result of the maximization (minimization) of $J(\mathbf{w}(t))$ is well known: $\mathbf{u}(t)$ must be the principal (minor) eigenvector of the 2×2 symmetric matrix $\underline{\mathbf{C}}(t)$. Thus, $\mathbf{w}(t)$ can be tracked by computing :

- An orthonormal basis $\underline{\mathbf{V}}(t)$ of the range space of $\mathbf{V}(t)$.
- The matrix $\underline{\mathbf{C}}(t) = \underline{\mathbf{V}}(t)^H\mathbf{C}(t)\underline{\mathbf{V}}(t)$.
- The principal (minor) eigenvector of $\underline{\mathbf{C}}(t)$ denoted by $\mathbf{u}(t)$.
- The principal (minor) eigenvector of the covariance matrix $\mathbf{C}(t)$ can be written as $\mathbf{w}(t) = \underline{\mathbf{V}}(t)\mathbf{u}(t)$.

3. FAST IMPLEMENTATION OF MYAST

Below, a fast implementation of MYAST is proposed whose global cost is of $O(n)$ flops per iteration. It can be decomposed into three steps: computation of $\underline{\mathbf{V}}(t)$ (section 3.1), computation of $\underline{\mathbf{C}}(t)$ (section 3.2), computation of $\mathbf{u}(t)$ and update of $\mathbf{w}(t)$ (section 3.3).

3.1. Computation of $\underline{\mathbf{V}}(t)$

This is done by Gram Schmidt orthogonalization of $\mathbf{V}(t)$. Define the scalar $y(t) = \mathbf{w}(t-1)^H\mathbf{x}(t)$ and let $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{w}(t-1)y(t)$. The $n \times 1$ vector $\mathbf{e}(t)$ is orthogonal to $\mathbf{w}(t-1)$ and hence the orthonormal matrix $\underline{\mathbf{V}}(t)$ is given by

$$\underline{\mathbf{V}}(t) = \left[\mathbf{w}(t-1), \frac{\mathbf{e}(t)}{\|\mathbf{e}(t)\|}\right]. \tag{7}$$

3.2. Computation of $\underline{\mathbf{C}}(t)$

Substituting equation (7) into equation (6) and after some straightforward manipulations we get

$$\begin{aligned}
\underline{\mathbf{C}}_{1,1}(t) &= \beta c_y(t-1) + |y(t)|^2 \\
\underline{\mathbf{C}}_{1,2}(t) &= \frac{\beta}{\|\mathbf{e}(t)\|^2}(y'(t) - c_y(t-1)y(t)) + y(t)\,\|\mathbf{e}(t)\| \\
\underline{\mathbf{C}}_{2,1}(t) &= \underline{\mathbf{C}}^*_{1,2}(t) \\
\underline{\mathbf{C}}_{2,2}(t) &= \frac{\beta}{\|\mathbf{e}(t)\|^2}(\mathbf{x}(t)^H\mathbf{x}'(t) - 2\mathcal{R}e(y'(t)^* y(t))) + \\
&\quad |y(t)|^2\, c_y(t-1) + \|\mathbf{e}(t)\|^2
\end{aligned}$$

where

$$c_y(t) = \mathbf{w}(t)^H\mathbf{C}(t)\mathbf{w}(t) \tag{8}$$

$\mathbf{x}'(t) = \mathbf{C}(t-1)\mathbf{x}(t)$ and [1] $y'(t) = \mathbf{w}(t-1)^H\mathbf{x}'(t)$. The notation a^* represents the complex conjugate of a and $\mathcal{R}e(a)$ the real part of a, while $\underline{\mathbf{C}}_{i,j}$ represents the (i,j)-th entry of matrix $\underline{\mathbf{C}}$.

3.3. Update of $\mathbf{w}(t)$

As mentioned in equation (4), the principal (minor) eigenvector $\mathbf{w}(t)$ is written as the product of $\underline{\mathbf{V}}(t)$ and $\mathbf{u}(t)$. the Last vector represents the principal (minor) eigenvector of matrix $\underline{\mathbf{C}}(t)$. Since $\underline{\mathbf{C}}(t)$ is a 2×2 matrix, one can compute this eigenvector explicitly according to :

$$\begin{aligned}
\bar{\mathbf{u}}(t) &= \begin{bmatrix} \underline{\mathbf{C}}_{2,2}(t) + \underline{\mathbf{C}}_{1,2}(t) - \lambda \\ \lambda - \underline{\mathbf{C}}_{1,1}(t) - \underline{\mathbf{C}}_{1,2}(t)^* \end{bmatrix} && (9) \\
\mathbf{u}(t) &= \frac{\bar{\mathbf{u}}(t)}{\|\bar{\mathbf{u}}(t)\|} && (10)
\end{aligned}$$

where $\lambda = \frac{\underline{\mathbf{C}}_{1,1}(t)+\underline{\mathbf{C}}_{2,2}(t)+\epsilon\sqrt{\Delta}}{2}$ is the eigenvalue of $\underline{\mathrm{C}}$(t), $\Delta = (\underline{\mathbf{C}}_{1,1}(t) - \underline{\mathbf{C}}_{2,2}(t))^2 + 4\left|\underline{\mathbf{C}}_{1,2}(t)\right|^2$ and $\epsilon = +1$ (resp. $\epsilon = -1$) for the principal (resp. minor) eigenvalue. The updating of equation (8) is then given by $c_y(t) = \mathbf{u}(t)^H\underline{\mathbf{C}}(t)\mathbf{u}(t)$.

Our new algorithm MYAST summarized in Table 1, costs approximately $15n$ flops per iteration.

4. STABILITY ANALYSIS

Here, we analyse the numerical stability of the YAST algorithm for minor subspace tracking [5]. Its pseudo-code is summarized in Table 2 (in this table $\mathbf{W}(t)$ is the $n \times r$ weight matrix estimate of the desired minor subspace). To examine the numerical stability, we focus

[1]The computation of vector $\mathbf{x}'(t) = \mathbf{C}(t-1)\mathbf{x}(t)$ is reduced from $O(n^2)$ to $9n$ by means of the technique described in [6], which exploits the shift invariance property of time-series covariance matrices.

$$\boxed{\begin{aligned}
&y(t) = \mathbf{w}(t-1)^H\mathbf{x}(t) \\
&\mathbf{x}'(t) = \mathbf{C}(t-1)\mathbf{x}(t) \\
&y'(t) = \mathbf{w}(t-1)^H\mathbf{x}'(t) \\
&\|\mathbf{e}(t)\| = \sqrt{\|\mathbf{x}(t)\|^2 - |y(t)|^2)} \\
&\underline{\mathbf{C}}_{1,1}(t) = \beta c_y(t-1) + |y(t)|^2 \\
&\underline{\mathbf{C}}_{1,2}(t) = \tfrac{\beta}{\|\mathbf{e}(t)\|^2}(y'(t) - c_y(t-1)y(t)) + y(t)\,\|\mathbf{e}(t)\| \\
&\underline{\mathbf{C}}_{2,1}(t) = \underline{\mathbf{C}}^*_{1,2}(t) \\
&\underline{\mathbf{C}}_{2,2}(t) = \tfrac{\beta}{\|\mathbf{e}(t)\|^2}(\mathbf{x}(t)^H\mathbf{x}'(t) - 2\mathcal{R}e(y'(t)^* y(t)) + \\
&|y(t)|^2\, c_y(t-1) + \|\mathbf{e}(t)\|^2 \\
&\Delta = (\underline{\mathbf{C}}_{1,1}(t) - \underline{\mathbf{C}}_{2,2}(t))^2 + 4\left|\underline{\mathbf{C}}_{1,2}(t)\right|^2 \\
&\lambda = \tfrac{\underline{\mathbf{C}}_{1,1}(t)+\underline{\mathbf{C}}_{2,2}(t)+\epsilon\sqrt{\Delta}}{2} \\
&\bar{\mathbf{u}}(t) = \begin{bmatrix} \underline{\mathbf{C}}_{2,2}(t) + \underline{\mathbf{C}}_{1,2}(t) - \lambda \\ \lambda - \underline{\mathbf{C}}_{1,1}(t) - \underline{\mathbf{C}}_{1,2}(t)^\star \end{bmatrix} \\
&\mathbf{u}(t) = \tfrac{\bar{\mathbf{u}}(t)}{\|\bar{\mathbf{u}}(t)\|} = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \\
&\mathbf{w}(t) = \underline{\mathbf{V}}(t)\mathbf{u}(t) = (u_1(t) - y(t)\tfrac{u_2(t)}{\|\mathbf{e}(t)\|})\mathbf{w}(t-1) + \tfrac{u_2(t)}{\|\mathbf{e}(t)\|}\mathbf{x}(t) \\
&c_y(t) = \mathbf{u}(t)^H\underline{\mathbf{C}}(t)\mathbf{u}(t)
\end{aligned}}$$

Table 1. MYAST algorithm.

on the deviation of the algorithm from orthonormality. Let us first consider the matrix $\mathbf{W}(t)^H\mathbf{W}(t)$ which, in the ideal case should be equal to the identity. Using the equations of Table 2, and after some straightforward manipulations we get $\mathbf{W}(t)^H\mathbf{W}(t) = \mathbf{A} + \mathbf{B} + \mathbf{B}^H$ where[2]

$$\mathbf{A} = \mathbf{W}^H\mathbf{W} + \mathbf{f}(1 + \tfrac{1}{\|\mathbf{e}\|^2}(-\|\mathbf{y}\|^2 + \mathbf{y}^H\mathbf{W}^H\mathbf{W}\mathbf{y}))\mathbf{f}^H - 2\mathbf{f}\mathbf{v}^H\mathbf{W}\mathbf{W}^H\mathbf{v}\mathbf{f}^H$$
$$+ \mathbf{f}\mathbf{f}'^H\mathbf{W}^H\mathbf{W}\mathbf{f}'\mathbf{f}^H + \mathbf{f}\mathbf{v}^H(\mathbf{W}\mathbf{W}^H)^2\mathbf{v}\mathbf{f}^H,$$

$$\mathbf{B} = \mathbf{f}\mathbf{v}^H\mathbf{W}\mathbf{f}'\mathbf{f}^H - \left(\mathbf{I} - \mathbf{W}^H\mathbf{W}\right)\mathbf{W}^H\mathbf{v}\mathbf{f}^H - \mathbf{f}\mathbf{f}'^H\mathbf{W}^H\mathbf{W}\mathbf{W}^H\mathbf{v}\mathbf{f}^H$$
$$- \mathbf{W}^H\mathbf{W}\mathbf{f}'\mathbf{f}^H.$$

Let $\boldsymbol{\epsilon}(t)$ be the deviation from orthonormality due to numerical rounding errors i.e.

$$\mathbf{W}(t)^H\mathbf{W}(t) = \mathbf{I} + \boldsymbol{\epsilon}(t). \tag{11}$$

Using this expression together with the expression of vector $\mathbf{v}(t)$ given in Table 2, leads to

$$\mathbf{W}(t-1)^H\mathbf{v}(t) = \frac{-1}{\|\mathbf{e}\|}\boldsymbol{\epsilon}(t-1)\mathbf{y}(t). \tag{12}$$

Now, by replacing (11) and (12) into the above expression of matrices $\mathbf{A}$ and $\mathbf{B}$ and keeping only the first order terms in $\boldsymbol{\epsilon}(t-1)$, we obtain

$$\begin{aligned}
\boldsymbol{\epsilon}(t) = \boldsymbol{\epsilon}(t-1) - \frac{1}{1+\rho}\mathbf{f}\mathbf{f}^H\boldsymbol{\epsilon}(t-1) - \frac{1}{1+\rho}\boldsymbol{\epsilon}(t-1)\mathbf{f}\mathbf{f}^H + \\
\frac{1}{(1+\rho)^2}\mathbf{f}\mathbf{f}^H\boldsymbol{\epsilon}(t-1)\mathbf{f}\mathbf{f}^H + \frac{1}{\|\mathbf{e}\|^2}\mathbf{f}\mathbf{y}^H\boldsymbol{\epsilon}(t-1)\mathbf{y}\mathbf{f}^H
\end{aligned} \tag{13}$$

or equivalently,

$$vec\left(\boldsymbol{\epsilon}(t)\right) = \mathbf{N}vec\left(\boldsymbol{\epsilon}(t-1)\right)$$

where $vec(.)$ represents the column vectorization operator and $\mathbf{N} = \mathbf{I}_{r^2} - \frac{1}{1+\rho}\mathbf{I}_r \otimes \mathbf{f}\mathbf{f}^H - \frac{1}{1+\rho}\mathbf{f}\mathbf{f}^H \otimes \mathbf{I}_r + \frac{1}{(1+\rho)^2}\mathbf{f}\mathbf{f}^H \otimes \mathbf{f}\mathbf{f}^H + \frac{1}{\|\mathbf{e}\|^2}\left(\mathbf{f}\mathbf{y}^H \otimes \mathbf{f}\mathbf{y}^H\right)$, where $\otimes$ is the matrix Kronecker product.
This expression leads to

$$\|\boldsymbol{\epsilon}(t)\|^2 = \|vec\left(\boldsymbol{\epsilon}(t)\right)\|^2 = vec\left(\boldsymbol{\epsilon}(t-1)\right)^H\mathbf{N}^H\mathbf{N}vec\left(\boldsymbol{\epsilon}(t-1)\right)$$

and hence the algorithm would be stable only if all the eigenvalues of $\mathbf{N}^H\mathbf{N}$ are strictly smaller than one [12]. This is unfortunately not

[2]$\mathbf{A}, \mathbf{B}$ are function of $\mathbf{W}(t-1)$ but for simplicity we omit to specify the time dependence.

$$
\begin{array}{|l|}
\hline
\mathbf{y}(t)=\mathbf{W}(t-1)^H\mathbf{x}(t)\\
\mathbf{x}^{'}(t)=\mathbf{C}(t-1)\mathbf{x}(t)\\
\mathbf{y}^{'}(t)=\mathbf{W}(t-1)^H\mathbf{x}^{'}(t)\\
\mathbf{e}(t)=\mathbf{x}(t)-\mathbf{W}(t-1)\mathbf{y}(t)\\
\|\mathbf{e}(t)\|=\sqrt{\|\mathbf{x}(t)\|^2-\|\mathbf{y}(t)\|^2}\\
\mathbf{v}(t)=\frac{1}{\|\mathbf{e}(t)\|}\mathbf{e}(t)\\
\alpha(t)=\|\mathbf{x}(t)\|^2\\
\mathbf{y}^{''}(t)=\beta\mathbf{y}^{'}(t)+\mathbf{y}(t)\,\|\mathbf{x}(t)\|^2\\
c_y(t)=\beta\mathbf{x}(t)^H\mathbf{x}^{'}(t)+\|\mathbf{x}(t)\|^4\\
\mathbf{C}_y^{'}(t)=\beta\mathbf{C}_y(t-1)+\mathbf{y}(t)\mathbf{y}(t)^H\\
\mathbf{h}(t)=\mathbf{y}^{''}(t)-\mathbf{C}_y^{'}(t)\mathbf{y}(t)\\
\gamma(t)=c_y(t)-\mathbf{y}(t)^H\mathbf{h}(t)-\mathbf{y}^{''}(t)^H\mathbf{y}(t)\\
\mathbf{g}(t)=\frac{-1}{\|\mathbf{e}(t)\|}\mathbf{h}(t)\\
\gamma^{'}(t)=\frac{\gamma(t)}{\|\mathbf{e}(t)\|^2}\\
\underline{\mathbf{C}}=\left[\mathbf{C}_y^{'}(t),-\mathbf{g}(t);-\mathbf{g}(t)^H,\gamma^{'}(t)\right]\\
(\boldsymbol{\phi}(t),\lambda(t))=\ eigs\ (\underline{\mathbf{C}}(t),1)\\
\left[\boldsymbol{\varphi}(t)^T,z(t)\right]=\boldsymbol{\phi}(t)^T\\
z(t)=\rho(t)e^{j\theta(t)}\ \text{(polar decomposition)}\\
\mathbf{f}(t)=\boldsymbol{\varphi}(t)\theta(t)\\
\mathbf{f}^{'}(t)=\frac{\mathbf{f}(t)}{1+\rho(t)}\\
\mathbf{e}^{'}(t)=\mathbf{v}(t)+\mathbf{W}(t-1)\left(\mathbf{f}^{'}(t)-\mathbf{W}(t-1)^H\mathbf{v}(t)\right)\\
\mathbf{W}(t)=\mathbf{W}(t-1)-\mathbf{e}^{'}(t)\mathbf{f}(t)^H\\
\mathbf{g}^{'}(t)=\mathbf{g}(t)+\mathbf{f}^{'}(t)\left(\gamma^{'}(t)-\theta^2(t)\lambda(t)\right)\\
\mathbf{C}_y(t)=\mathbf{C}_y^{'}(t)+\mathbf{g}^{'}(t)\mathbf{f}^{'}(t)^H+\mathbf{f}^{'}(t)\mathbf{f}(t)^H\\
\hline
\end{array}
$$

Table 2. YAST algorithm.

the case here as shown by the following Lemma :

Lemma 1 : The YAST algorithm is numerically instable as matrix $\mathbf{N}^H\mathbf{N}$ *has eigenvalues larger than* 1.

Proof : Let us consider a unitary $r\times 1$ vector $\mathbf{a}$ orthogonal to $\mathbf{f}$, we have:

$$\mathbf{N}\left(\mathbf{a}\otimes\mathbf{a}\right)=\mathbf{a}\otimes\mathbf{a}+\frac{(\mathbf{y}^H\mathbf{a})^2}{\|\mathbf{e}\|^2}\mathbf{f}\otimes\mathbf{f}.$$

These two vectors, being orthogonal, we have then

$$
\begin{aligned}
\|\mathbf{N}\left(\mathbf{a}\otimes\mathbf{a}\right)\|^2 &= \|\mathbf{a}\otimes\mathbf{a}\|^2+\left|\frac{\mathbf{y}^H\mathbf{a}}{\|\mathbf{e}\|}\right|^4\|\mathbf{f}\otimes\mathbf{f}\|^2\\
&= 1+\left|\frac{\mathbf{y}^H\mathbf{a}}{\|\mathbf{e}\|}\right|^4\|\mathbf{f}\|^4
\end{aligned}
$$

which is clearly larger than one. Hence, the deviation from orthonormality does increase at each iteration which means that the YAST algorithm for minor component extraction is numerically instable.

5. YAST-PGS

To mitigate this effect, we propose to use an efficient partial orthogonalization scheme called Pairwise Gram-Schmidt (PGS) orthogonalization [11] so that we refer to this algorithm version by YAST-PGS. The PGS consists simply in the re-orthogonalization of two successive column vectors of $\mathbf{W}$ at each iteration, *i.e.* if $\mathbf{W}(t)\triangleq\left[\begin{array}{ccc}\mathbf{w}_0(t) & \dots & \mathbf{w}_{r-1}(t)\end{array}\right]$ represents the weight matrix at iteration t, we propose to choose two column vectors of index $k_t = mod_r(t)$ and $k_{t+1} = mod_r(t+1)$, where mod_r is the modulo r value, and perform their Gram-Schmidt orthogonalization according to Table 3. The Yast-PGS algorithm can be resumed as in Table 2 with the difference that, in the expression of vector $\mathbf{e}^{'}(t)$, we do

$$
\begin{array}{|lcl|}
\hline
\mathbf{w}_{k_{t+1}}(t) & := & \mathbf{w}_{k_{t+1}}(t)-\mathbf{w}_{k_t}(t)\mathbf{w}_{k_t}^H(t)\mathbf{w}_{k_{t+1}}(t)\\
\mathbf{w}_{k_{t+1}}(t) & := & \frac{\mathbf{w}_{k_{t+1}}(t)}{\|\mathbf{w}_{k_{t+1}}(t)\|}\\
\hline
\end{array}
$$

Table 3. YAST-PGS Algorithm

not project $\mathbf{v}(t)$ onto the signal subspace (i.e. in Table 2, we have $\mathbf{e}^{'}(t)=(\mathbf{I}-\mathbf{W}(t-1)\mathbf{W}(t-1)^H)\mathbf{v}(t)+\mathbf{W}(t-1)\mathbf{f}^{'}(t)$ where the subspace projector $(\mathbf{I}-\mathbf{W}(t-1)\mathbf{W}(t-1)^H)$ is introduced only for the algorithm stabilization, since, in the absence of numerical rounding errors, we have $(\mathbf{I}-\mathbf{W}(t-1)\mathbf{W}(t-1)^H)\mathbf{v}(t)=\mathbf{v}(t)$) but rather we compute $\mathbf{e}^{'}(t)=\mathbf{v}(t)+\mathbf{W}(t-1)\mathbf{f}^{'}(t)$. This reduces the computational cost by nr flops per iteration. Thanks to this re-orthogonalization and re-normalization, the accumulated numerical error becomes stable as shown in figures 3 and 4.

6. SIMULATIONS

To assess the performance of our algorithm, we calculate the ensemble average of the performance factor $\rho(t)$ expressed in the case $r=1$ by

$$\rho(t)=\frac{1}{p_0}\sum_{p=1}^{p_0}\|\mathbf{W}_p(t)-\mathbf{E}_2\|^2$$

and in the case $r>1$ by

$$\rho(t)=\frac{1}{p_0}\sum_{p=1}^{p_0}\frac{tr(\mathbf{W}_p^H(t)\mathbf{E}_1\mathbf{E}_1^H\mathbf{W}_p(t))}{tr(\mathbf{W}_p^H(t)\mathbf{E}_2\mathbf{E}_2^H\mathbf{W}_p(t))}$$

where the number of algorithm runs is $p_0=100$, p indicates that the associated variable depends on the particular run. $\mathbf{E_2}$ is the $n\times r$ matrix of the r minor (or principal if we target the PSA) eigenvectors and $\mathbf{E_1}$ is the $n\times n-r$ matrix of the principal eigenvectors. This performance index ρ measures the averaged estimation quality of the target subspace. To measure the deviation from the orthonormality, we use the performance index

$$\eta(i)=\frac{1}{p_0}\sum_{p=1}^{p_0}\|\mathbf{W}_p^H(t)\mathbf{W}_p(t)-\mathbf{I}\|_F^2$$

In the simulation experiment, we have considered an iid sequence of n-dimensional random vectors $\mathbf{x}(t)$. For figures 1, 2 and 3 ($n=4$), the random sequence is generated using a zero mean Gaussian-distribution with the well known [5] covariance matrix $\mathbf{C}=\left[\begin{array}{cccc}0.9 & 0.4 & 0.7 & 0.3\\ 0.4 & 0.3 & 0.5 & 0.4\\ 0.7 & 0.5 & 1.0 & 0.6\\ 0.3 & 0.4 & 0.6 & 0.9\end{array}\right]$. In Fig.1, we extract the principal eigenvectors of $\mathbf{C}$ using the proposed MYAST but also for comparison, the PASTd method [8] and the singular value decomposition (SVD) applied to the updated covariance matrix $\mathbf{C}(t)=\beta\mathbf{C}(t-1)+\mathbf{x}(t)\mathbf{x}(t)^H$. One can observe that PASTd and MYAST have the same behavior and reach the performance of the SVD method.

In Fig.2, a similar experiment is done but this time to extract the minor eigenvector of $\mathbf{C}$. One observes that MYAST provides better estimation performance than PASTd but now its convergence rate is lower than that of the SVD.

In Fig.3, we consider the MSA case with $r=2$ and compare the performance of the YAST, YAST-PGS and SVD. To highlight the effect of the numerical errors, we have used a rounding precision of 10 digits. One can observe that in this context, YAST-PGS behaves as well as the SVD and has a very good stability behavior. As we can observe, this is not the case for YAST which is numerically instable due to the degeneration of the weight matrix in the sense that its column vectors become almost linearly dependent and close to the least

eigenvectors. For this reason, the value of ρ becomes lower than that reached by the SVD and MYAST-PGS.
For the least experiments, we choose $n = 15$ and generate $\mathbf{x}(t)$ as an iid sequence with a positive definite covariance matrix $\mathbf{C}$ that is generated randomly at each run. In Fig.4, we compare again YAST, YAST-PGS and SVD algorithms for the MSA with $r = 6$. As before, YAST-PGS is observed to be stable (but not the YAST) and leads to a better steady state estimation error than the SVD.

7. CONCLUSION

In this paper, we proposed first a simple modified version of the YAST algorithm (the MYAST) for the principal and minor eigenvector extraction. The MYAST is easy to implement and it reaches the performance of the expensive SVD algorithm. Then, we proved theoretically the numerical instability of the YAST algorithm and we proposed a stabilized version using the PGS technique. The resulting algorithm becomes more stable than YAST and has a lower computational cost.

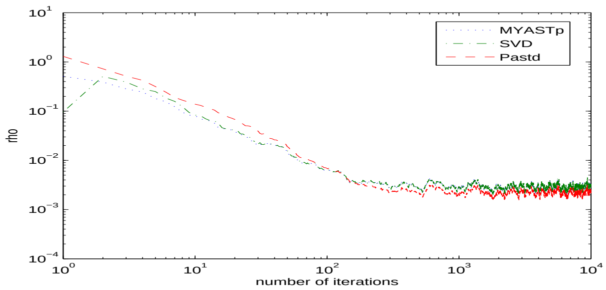

Fig. 1. Principal vector estimation with $(r = 1, n = 4)$

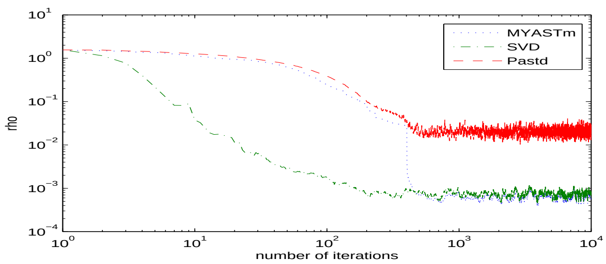

Fig. 2. Minor vector estimation with $(r = 1, n = 4)$

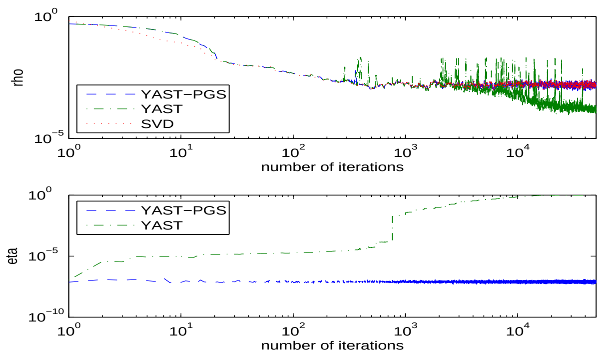

Fig. 3. MSA with $(r = 2, n = 4)$

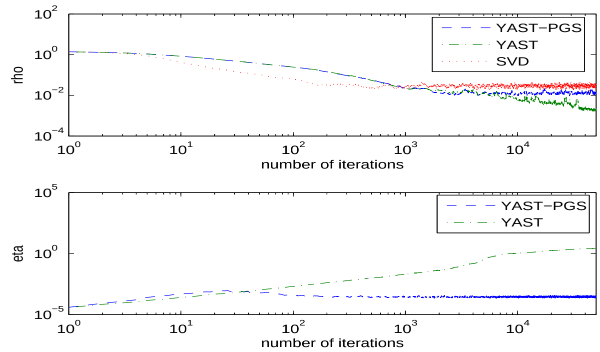

Fig. 4. MSA with $(r = 6, n = 15)$

8. REFERENCES

[1] X. Wang, H. V. Poor, P. Van Dooren and J. Vandewalle, "Blind Multiuser Detection: A Subspace Approach",*IEEE Transactions on Inf. Theory*, vol. 44 No. 2, pp. 677-689, March 1998.

[2] S. Bartelmaos, K. Abed-Meraim and S. Attallah,"Mobile localization using subspace tracking", *Asia-Pacific Conference on Communications",pp. 1009-1013, October 2005.*

[3] *I. Kacha, K. Abed-Meraim and A. Belouchrani, "Fast Adaptive Blind Equalization Method without Channel Order Estimation"*,ICASSP'06, *May 2006.*

[4] *Badeau R., David B. and Richard G., "yet another subspace tracker"*, Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05),*vol. 4, pp. 329-332, March 2005.*

[5] *R. Badeau, B. David and G. Richard, "YAST Algorithm for Minor Subspace Tracking"*, ICASSP 2006, *May 2006.*

[6] *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.*

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

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

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

[10] *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.*

[11] *M. Moonen, P. van Dooren, and J. Vandewalle, "A Note on Efficient Numericaly Stabilized Rank-One Eigenstructure Updating"*, IEEE Transactions on Signal Processing,*vol. 39 No. 8 pp. 1911-1914, Aug. 1991.*

[12] *X. G. Doukopoulos, "Power Techniques for Blind Channel Estimation in Wireless Communication Systems"*, PhD thesis, IRISA-INRIA, University of Rennes 1, France, *2004.*