Low complexity adaptive algorithms for Principal and Minor Component Analysis

Messaoud Thameri, Karim Abed-Meraim, Adel Belouchrani

Digital Signal Processing ••• (••••) •••–•••

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Low complexity adaptive algorithms for Principal and Minor Component Analysis

Messaoud Thameri [a,*], Karim Abed-Meraim [b], Adel Belouchrani [c]

[a] TELECOM ParisTech, TSI Department, France
[b] PRISME Laboratory, Polytech'Orléans, Orléans University, France
[c] EE Department, Ecole Nationale Polytechnique, Algeria

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
PCA
MCA
MSA
OPAST
Givens rotations
Data whitening
Adaptive algorithm

ABSTRACT

This article introduces new low cost algorithms for the adaptive estimation and tracking of principal and minor components. The proposed algorithms are based on the well-known OPAST method which is adapted and extended in order to achieve the desired MCA or PCA (Minor or Principal Component Analysis). For the PCA case, we propose efficient solutions using Givens rotations to estimate the principal components out of the weight matrix given by OPAST method. These solutions are then extended to the MCA case by using a transformed data covariance matrix in such a way the desired minor components are obtained from the PCA of the new (transformed) matrix. Finally, as a byproduct of our PCA algorithm, we propose a fast adaptive algorithm for data whitening that is shown to overcome the recently proposed RLS-based whitening method.

1. Introduction

Principal (resp. Minor) Component Analysis PCA (resp. MCA) is a major problem in many practical applications including direction of arrival estimation [1], system identification [2–4], spectral analysis [5], multiuser detection in communications [6,7] and space–time adaptive filtering for radar systems [8].

Unlike Principal (resp. Minor) Subspace Analysis PSA (resp. MSA), where only a basis of the principal (resp. minor) subspace is considered, PCA (resp. MCA) technique consists rather of estimating the eigenvectors of the data covariance matrix. While the conventional matrix algebraic approaches such as Singular Value Decomposition (SVD) provide a good solution for small system dimensions, they quickly become inappropriate (too expensive) for real time applications when the dimensions increase.

In the literature, PCA and MCA methods can be grouped in many categories [9,10]:

- According to their processing methodology: sequential or parallel versions:
  - In the sequential version [11–13], the desired eigenvectors are extracted sequentially. This processing approach may lead to an important processing delay and error propagation effect.
  - Parallel structures [14–17] extract jointly all desired components. This avoids the previously aforementioned problems but may lead to more expensive implementations and to numerical instability.
- According to their numerical complexities: low complexity $O(np)$, moderate complexity $O(np^2)$ or $O(n^2)$ and high complexity $O(n^2p)$ or more, where n is the size of the observation vector and $p < n$ is the number of eigenvectors to estimate.
  - Most low complexity algorithms [12,11,18–21] are based on gradient techniques. Unfortunately, many of these algorithms suffer from low convergence rate or numerical instability.
  - Moderate complexity $O(np^2)$ or $O(n^2)$ is due to the orthonormalization step of the estimate eigenvectors [22,23] or to some $n \times n$ matrix-vector products used in the PCA or MCA updating [24,25]. Moderate complexity algorithms update only the desired eigen-structure and present a good trade-off between numerical complexity and performance (i.e. fast convergence rate, orthonormality of the weight matrix, good stability, ...).
  - Unlike moderate complexity algorithms, those of high complexity, i.e. $O(n^2p)$ or more, correspond to methods that update the complete eigen-structure with or without explicit computation of the covariance matrix [14].

[*] Corresponding author.
E-mail addresses: thameri@telecom-paristech.fr (M. Thameri), karim.abed-meraim@univ-orleans.fr (K. Abed-Meraim), adel.belouchrani@enp.edu.dz (A. Belouchrani).

1051-2004/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.dsp.2012.09.007

In this paper, we propose algorithms for PCA and MCA problems with low complexity and improved convergence performance.[1] These algorithms are based on the Orthogonal Projection Approximation and Subspace Tracking (OPAST) algorithm [27], originally introduced for Principal Subspace Analysis (PSA). In [28], authors propose to use jointly OPAST algorithm and a diagonalization technique using Givens rotations to achieve the PCA. The resulting algorithm shows good performance but suffers from ill convergence when the system's dimensions increase or the number of principal components is large. To improve its performance in the large dimensional context, we propose herein different algorithm's versions using different selection procedures of the Givens rotation indices. A comparative study shows that the best algorithm's version (in terms of convergence rate and estimation accuracy) is the one associated to the hybrid selection method shown in Section 3.2.1.

On the other hand, we propose two algorithms for MCA referred to as α-GOPAST1[2] and α-GOPAST2 respectively. These algorithms use the transformed covariance matrix $\mathbf{C}'_{xx} = \alpha\mathbf{I} - \mathbf{C}_{xx}$ where α is a properly chosen scalar factor and $\mathbf{C}_{xx}$ is the covariance matrix of the observed data. This transformation allows us to compute the minor eigenvectors of $\mathbf{C}_{xx}$ as the principal eigenvectors of $\mathbf{C}'_{xx}$. We propose 2 algorithm's versions depending on the way the transformed covariance matrix and the desired weight matrix are updated. For both algorithms, fast implementations are proposed and analyzed through numerical simulation.

Finally, because of the importance of data whitening in certain practical applications, e.g. enhanced direction of arrival by pre-whitening [29] and blind source separation [30], we have proposed an adaptive data whitening algorithm based on our PCA method. This algorithm shows excellent convergence rate and better 'whitening quality' as compared to the existing methods.

This paper is organized as follows: Objectives, data model and optimization criterion are given in Section 2. Section 3 presents the Generalized OPAST (denoted GOPAST) and our contributions to improve its convergence performance. The whitening algorithm is presented in Section 4 while in Section 5, we propose different algorithms to solve MSA and MCA problems. Section 6 shows the numerical results and finally the main conclusions are summarized in Section 7.

2. Problem formulation

2.1. Data model and objectives

Let $\mathbf{x}(t) \in \mathbb{C}^{n\times 1}$ be the observed data vector of covariance matrix $\mathbf{C}_{xx}$. We aim to estimate and track the $p < n$ principal (resp. minor) eigenvectors of $\mathbf{C}_{xx}$. For that, we assume that the sorted eigenvalues of $\mathbf{C}_{xx}$ satisfy the conditions:

(H1) for the PCA, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p > \lambda_{p+1} \geqslant \cdots \geqslant \lambda_n$

(H2) for the MCA, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_{n-p} > \lambda_{n-p+1} \geqslant \cdots \geqslant \lambda_n$

In the adaptive scheme, matrix $\mathbf{C}_{xx}$ is evaluated at time instant t as[3]

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

where $0 < \beta < 1$ is a chosen forgetting factor. Its eigenvalues/eigenvectors are then evaluated by applying fast adaptive optimization techniques to the cost function described below.

Table 1
OPAST algorithm.

```
Initialization: $\mathbf{W}(0) = \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0} \end{bmatrix}$, $\mathbf{Z}(0) = \mathbf{I}_p$
Adaptation: For $t \geqslant 1$
  $\mathbf{y} = \mathbf{W}^H(t-1)\mathbf{x}(t)$
  $\mathbf{q} = \frac{1}{\beta}\mathbf{Z}(t-1)\mathbf{y}$
  $\gamma = \frac{1}{1+\mathbf{y}^H\mathbf{q}}$
  $\tau = \frac{1}{\|\mathbf{q}\|^2}\left(\frac{1}{\sqrt{1+\|\mathbf{q}\|^2\gamma^2(\|\mathbf{x}(t)\|^2-\|\mathbf{y}\|^2)}} - 1\right)$
  $\mathbf{e} = \mathbf{W}(t-1)(\tau\mathbf{q} - \gamma(1+\tau\|q\|^2)\mathbf{y}) + (1+\tau\|q\|^2)\gamma\mathbf{x}(t)$
  $\mathbf{Z}(t) = \frac{1}{\beta}\mathbf{Z}(t-1) - \gamma\mathbf{q}\mathbf{q}^H$
  $\mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{e}\mathbf{q}^H$
```

2.2. Optimization criterion

Consider the scalar function

$$\begin{aligned} J(\mathbf{W}) &= \mathbf{E}\left(\left\|\mathbf{x}(t) - \mathbf{W}\mathbf{W}^H\mathbf{x}(t)\right\|^2\right) \\ &= \operatorname{trace}\left(\mathbf{C}_{xx} - 2\mathbf{W}^H\mathbf{C}_{xx}\mathbf{W} + \mathbf{W}^H\mathbf{C}_{xx}\mathbf{W}\mathbf{W}^H\mathbf{W}\right) \end{aligned} \tag{2}$$

where $\mathbf{W} \in \mathbb{C}^{n\times p}$ is a given unitary matrix, i.e., $\mathbf{W}^H\mathbf{W} = \mathbf{I}$.

It is shown in [12] that the optimization of J under unitary constraint leads to the following results.

- $\mathbf{W}$ is a stationary point of $J(\mathbf{W})$ if and only if $\mathbf{W} = \mathbf{V}_p\mathbf{Q}$, where $\mathbf{V}_p$ is an $n \times p$ matrix containing any p distinct eigenvectors of $\mathbf{C}_{xx}$, and $\mathbf{Q}$ is any $p \times p$ unitary matrix.
- All stationary points of $J(\mathbf{W})$ are saddle points, except when $\mathbf{V}_p$ contains the p-dominant eigenvectors (resp. $\mathbf{V}_p$ contains the p minor eigenvectors) of $\mathbf{C}_{xx}$, in which case $J(\mathbf{W})$ attains its minimum (resp. attains its maximum).

3. PCA algorithms

GOPAST algorithm consists of using the OPAST algorithm to compute the principal subspace weight matrix. Then a diagonalization technique using Givens rotations extracts the principal components from the weight matrix computed by OPAST. The former algorithm is described in the next subsection while the Givens rotation-based diagonalization technique is detailed in Section 3.2.

3.1. OPAST

Minimizing (2) iteratively leads to the following abstract form [12] of the PAST method

$$\mathbf{W}(t) = \mathbf{C}_{xx}(t)\mathbf{W}(t-1)\left(\mathbf{W}^H(t-1)\mathbf{C}_{xx}(t)\mathbf{W}(t-1)\right)^{-1} \tag{3}$$

OPAST algorithm consists of (3) followed by a fast orthonormalization of the weight matrix $\mathbf{W}(t)$ [27,31]. The fast implementation of (3) is based on the projection approximation $\mathbf{C}_{xx}(t)\mathbf{W}(t) \approx \mathbf{C}_{xx}(t)\mathbf{W}(t-1)$ which is clearly valid if the weight matrix $\mathbf{W}(t)$ is slowly time varying. With this approximation, the matrix product $\mathbf{C}_{xx}(t)\mathbf{W}(t-1)$, as well as the matrix inverse $\mathbf{Z}(t) = (\mathbf{W}^H(t-1)\mathbf{C}_{xx}(t)\mathbf{W}(t-1))^{-1}$, can be computed in $O(np)$ flops (see [12] for more details). OPAST algorithm is summarized in Table 1, showing clearly that its numerical complexity is $3np + p^2 + n + O(p)$ flops per iteration (which is classified in the linear complexity category).

[1] Part of this work related to PCA has been published in the conference paper [26].

[2] G stands for Generalized while α refers to the scalar coefficient used in the transformed covariance matrix $\mathbf{C}'_{xx} = \alpha\mathbf{I} - \mathbf{C}_{xx}$.

[3] This is the standard updating form of the covariance matrix using an exponential window (see [12, Eq. (15)]), i.e., $\mathbf{C}_{xx}(t) = (1-\beta)\sum_{i=1}^{t}\beta^{t-i}\mathbf{x}(i)\mathbf{x}(i)^H$.

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

3.2. Givens rotation-based diagonalization technique

Once the weight matrix of principal subspace $\mathbf{W}(t)$ is computed by OPAST algorithm, we extract the principal components using Givens rotations at each iteration. Indeed, as mentioned above, $\mathbf{W}(t)$ can be written as $\mathbf{W}(t) \approx \mathbf{V}_p(t)\mathbf{Q}(t)$, where $\mathbf{V}_p(t)$ is the matrix of the p-dominant eigenvectors of $\mathbf{C}_{xx}(t)$ and $\mathbf{Q}(t)$ is a given unitary matrix. Therefore, if the matrix $\mathbf{Q}(t)$ is known, the desired eigenvectors matrix $\mathbf{V}_p(t)$ can be obtained as $\mathbf{V}_p(t) = \mathbf{W}(t)\mathbf{Q}^H(t)$. Now, to determine the unknown matrix $\mathbf{Q}(t)$, we observe that, if $\mathbf{W} = \mathbf{V}_p\mathbf{Q}$, then matrix $\mathbf{Z} = (\mathbf{W}^H\mathbf{C}_{xx}\mathbf{W})^{-1} = \mathbf{Q}^H\mathbf{\Lambda}_p^{-1}\mathbf{Q}$, where $\mathbf{\Lambda}_p$ is the $p \times p$ diagonal matrix of principal eigenvalues of $\mathbf{C}_{xx}$. Hence, at time instant t, we have $\mathbf{Z}(t) \approx \mathbf{Q}^H(t)\mathbf{\Lambda}_p^{-1}(t)\mathbf{Q}(t)$ showing that $\mathbf{Q}(t)$ can be computed as the unitary matrix that diagonalizes $\mathbf{Z}(t)$.

It is known [32] that any $p \times p$ unitary matrix can be decomposed into a product of elementary Givens rotations

$$\mathbf{Q} = \prod_{\#sweeps} \prod_{1 \leqslant l < m \leqslant p} \mathbf{\Theta}_{lm} \tag{4}$$

where *#sweeps* stands for the number of sweeps, each sweep represents the processing of all $p(p-1)/2$ pairs of indices (l, m). The elementary Givens rotations $\mathbf{\Theta}_{lm}$ are defined as unitary matrices where all diagonal elements are 1 except for the two elements equal to $c = \cos(\theta)$ in rows (and columns) l and m. Likewise, all off-diagonal elements of $\mathbf{\Theta}_{lm}$ are 0 except for the two elements $s = \sin(\theta)\exp(\jmath\beta)$ and $-\bar{s}$ at positions (l, m) and (m, l), respectively, where $\bar{s}$ denotes the conjugate of s.

The considered algorithm consists of multiplying iteratively $\mathbf{Z}$ at the left and the right sides by an elementary Givens rotation and its transconjugate ($\mathbf{Z}' = \mathbf{\Theta}_{lm}\mathbf{Z}\mathbf{\Theta}_{lm}^H$) in such a way the transformed matrix is getting 'closer' to diagonal structure at each iteration. Similarly, the weight matrix $\mathbf{W}$ is iteratively multiplied at the right by the transconjugate of the Givens rotation: $\mathbf{W}' = \mathbf{W}\mathbf{\Theta}_{lm}^H$.

Now, to achieve the previously mentioned diagonalization, one needs to specify how the rotation indices are chosen at each iteration as well as how the angle parameters (θ, β) of Givens rotation are optimized. Below, we start by proposing different selection strategies for the rotation indices before giving details on the optimal computation of the rotation parameters.

3.2.1. Rotation indices selection

We introduce here four methods for the rotation indices selection. These methods are latter compared and discussed via numerical experiments. These selection methods are performed according to the different criteria shown below:

Maximum error deviation criterion: Givens rotation indices l and m are selected as[4]

$$(l, m) = \arg\max_{i \neq j} \left|\mathbf{Z}_{ij}(t)\right| \tag{5}$$

where $\mathbf{Z}_{ij}(t)$ is the (i, j)th entry of $\mathbf{Z}(t)$. As we can see, l and m are the indices of the off-diagonal element of $\mathbf{Z}(t)$ which deviates the most from zero (i.e. from the desired diagonal structure of matrix $\mathbf{Z}(t)$). Hence, this selection is referred to as "maximum error deviation criterion". This selection procedure has been originally proposed in [28]. The resulting algorithm's version is referred to as Maximum Error Deviation GOPAST (MED-GOPAST).

Table 2
Givens rotations-based diagonalization algorithm.

```
Choose (l, m) according to the proposed rotation indices selection procedures
$\mathbf{g} = [Z_{ll}(t) - Z_{mm}(t), 2\Re(Z_{lm}(t)), 2\Im(Z_{lm}(t))]^T$
$\mathbf{v} = \mathbf{g}/\|\mathbf{g}\| \operatorname{sign}(g_1)$
$c = \sqrt{\frac{v_1+1}{2}}$ and $s = \frac{v_2 + \jmath v_3}{2c}$
$\mathbf{Z}'(t) = \mathbf{Z}(t)$
$\mathbf{Z}'_{:,l}(t) = c\mathbf{Z}_{:,l}(t) + \bar{s}\mathbf{Z}_{:,m}(t)$
$\mathbf{Z}'_{:,m}(t) = c\mathbf{Z}_{:,m}(t) - s\mathbf{Z}_{:,l}(t)$
$\mathbf{Z}'_{l,:}(t) = c\mathbf{Z}_{l,:}(t) + s\mathbf{Z}_{m,:}(t)$
$\mathbf{Z}'_{m,:}(t) = c\mathbf{Z}_{m,:}(t) - \bar{s}\mathbf{Z}_{l,:}(t)$
$\mathbf{Z}(t) = \mathbf{Z}'(t)$
$\mathbf{W}'(t) = \mathbf{W}(t)$
$\mathbf{W}'_{:,l}(t) = c\mathbf{W}_{:,l}(t) + \bar{s}\mathbf{W}_{:,m}(t)$
$\mathbf{W}'_{:,m}(t) = c\mathbf{W}_{:,m}(t) - s\mathbf{W}_{:,l}(t)$
$\mathbf{W}(t) = \mathbf{W}'(t)$
```

Improved maximum error deviation criterion: For a Givens rotation $\mathbf{\Theta}_{lm}$, it is shown in Section 3.2.2 that the maximum relative diagonalization criterion increment is given by $\frac{2|\mathbf{Z}_{lm}|^2}{|\mathbf{Z}_{ll}|^2+|\mathbf{Z}_{mm}|^2}$.

We propose to use this criterion for the selection of the rotation indices according to

$$(l, m) = \arg\max_{i \neq j} \frac{|\mathbf{Z}_{ij}(t)|^2}{|\mathbf{Z}_{ii}(t)|^2 + |\mathbf{Z}_{jj}(t)|^2} \tag{6}$$

This criterion can be seen as an improved[5] version of the selection cost function in (5) and consequently we denote it IMED criterion and the corresponding algorithm's version IMED-GOPAST.

Automatic selection criterion: This criterion consists of choosing Givens rotation indices (l, m) according to an automatic selection (i.e. automatic incrementation) throughout the iterations in such a way all search directions (i.e. all indices values) are visited periodically. Hence, if (l, m) are the rotation indices at time instant $t-1$, then at the current time instant, we'll have

$$(l', m') = \begin{cases} (l, m+1) & \text{if } m < p \\ (l+1, l+2) & \text{if } m = p \text{ and } l < p-1 \\ (1, 2) & \text{if } m = p \text{ and } l = p-1 \end{cases} \tag{7}$$

This selection leads to a good estimation accuracy in the simulation results that can be explained by the fact that it allows us to better avoid the local minima. This algorithm's version is referred to as Automatic Selection GOPAST (AS-GOPAST).

Hybrid criterion: It consists of combining the first and the last selection criteria, i.e. we propose to use 2 rotations per time instant: In the first rotation, the Givens rotation indices are chosen according to the maximum error deviation criterion (5). Then, if (l, m) are the Givens rotation indices of the second rotation at time instant $t-1$, the indices of the second rotation at the current iteration are computed according to (7). Note that if the new indices are the same as those of the first rotation at time t, we have to increment them again according to (7). The algorithm's version using this selection procedure, referred to as Hybrid GOPAST (H-GOPAST), is shown to present a good trade-off between the convergence rate of MED-GOPAST and the estimation accuracy of AS-GOPAST.

[4] For simplicity, we still use the notation $\mathbf{Z}(t)$ to represent the transformed matrix using Givens rotation, i.e. after updating, we set $\mathbf{Z}' = \mathbf{Z}$ as shown in Table 2.

[5] Indeed, this criterion is directly related to diagonalization criterion increment and hence it improves, as show in Section 6, the algorithm's convergence performance.

Remark. In this work, we have chosen to use one single (or two for the hybrid method) Givens rotation per time instant. Obviously, one can choose to use several Givens rotations per time instant. However, increasing the number of Givens rotations will increase the numerical cost but it does not increase significantly H-GOPAST's performance as shown by our simulation results.

3.2.2. *Rotation parameters optimization*

Once the Givens rotation indices are obtained by one of the proposed selection procedures, we compute the rotation angles of $\mathbf{\Theta}_{lm}$ that minimize the sum of the square modulus of the off-diagonal entries of $\mathbf{Z}' = \mathbf{\Theta}_{lm}\mathbf{Z}\mathbf{\Theta}_{lm}^H$

$$(\theta, \beta) = \arg\min_{\theta,\beta} \sum_{a \neq b} \left|\mathbf{Z}'_{ab}(t)\right|^2$$

which can be shown to be equivalent to

$$(\theta, \beta) = \arg\max_{\theta,\beta} \left(\left|\mathbf{Z}'_{ll}(t)\right|^2 + \left|\mathbf{Z}'_{mm}(t)\right|^2\right) \tag{8}$$

Eq. (8) is referred to as diagonalization criterion. A direct computation of $\mathbf{Z}'_{ll}(t)$ and $\mathbf{Z}'_{mm}(t)$ leads to

$$\mathbf{Z}'_{ll}(t) = c^2\mathbf{Z}_{ll}(t) + \left|s^2\right|\mathbf{Z}_{mm}(t) + \bar{s}c\mathbf{Z}_{lm}(t) + sc\mathbf{Z}_{ml}(t)$$

$$\mathbf{Z}'_{mm}(t) = c^2\mathbf{Z}_{mm}(t) + \left|s^2\right|\mathbf{Z}_{ll}(t) - \bar{s}c\mathbf{Z}_{lm}(t) - sc\mathbf{Z}_{ml}(t)$$

Using the equalities $c^2 = (\cos(2\theta) + 1)/2$, $|s^2| = (1 - \cos(2\theta))/2$, and $cs = \sin(2\theta)e^{j\beta}/2$, we obtain for (8)

$$\begin{aligned}(\theta, \beta) &= \arg\max_{\theta,\beta}\Big[\left|\mathbf{v}^T\mathbf{g}_{lm}(t) + \mathbf{Z}_{ll}(t) + \mathbf{Z}_{mm}(t)\right|^2 \\ &\qquad + \left|-\mathbf{v}^T\mathbf{g}_{lm}(t) + \mathbf{Z}_{ll}(t) + \mathbf{Z}_{mm}(t)\right|^2\Big] \\ (\theta, \beta) &= \arg\max_{\theta,\beta}\left|\mathbf{v}^T\mathbf{g}_{lm}(t)\right|^2 \end{aligned} \tag{9}$$

where $\mathbf{v}$ is the unitary vector given by:

$$\mathbf{v} = \begin{bmatrix} \cos(2\theta) \\ \sin(2\theta)\cos(\beta) \\ \sin(2\theta)\sin(\beta) \end{bmatrix} \quad \text{and} \quad \mathbf{g}_{lm}(t) = \begin{bmatrix} Z_{ll}(t) - Z_{mm}(t) \\ 2\Re(Z_{lm}(t)) \\ 2\Im(Z_{lm}(t)) \end{bmatrix} \tag{10}$$

$\Re(\cdot)$ and $\Im(\cdot)$ being the real and imaginary parts of a complex entity. The optimal solution is given by $\mathbf{v} = \mathbf{g}_{lm}(t)/\|\mathbf{g}_{lm}(t)\|$, and the Givens rotation parameters c and s are calculated as

$$c = \sqrt{\frac{v_1 + 1}{2}}, \qquad s = \frac{v_2 + jv_3}{2c}$$

It is easy to check that this optimal solution corresponds to a relative increment of criterion (8) equal to $\frac{2|\mathbf{Z}_{lm}|^2}{|\mathbf{Z}_{ll}|^2 + |\mathbf{Z}_{mm}|^2}$.

Note that right multiplication by Givens rotation $\mathbf{\Theta}_{lm}^H$ leaves all column vectors unchanged except for the lth and mth column vectors that are changed according to the equations of Table 2. Similarly, left multiplication by Givens rotation $\mathbf{\Theta}_{lm}$ leaves all rows unchanged except for the lth and mth ones as shown in Table 2.

Remarks.

- Note that if $\mathbf{v}$ maximizes (9), then $-\mathbf{v}$ is also a maximum argument point. In our implementation, we have chosen the solution that has a positive-valued first entry. This is motivated by the projection approximation condition which assumes that $\mathbf{W}(t) \approx \mathbf{W}(t-1)$. Hence, we select the 'smallest' rotation angle that maximizes (9), i.e. among the two solutions, we choose the Givens rotation that is the 'closest' to identity matrix. This choice is important for the algorithm's convergence and avoids in particular the permutation of the columns of $\mathbf{W}(t)$ along the iterations.
- Givens rotations provide an additional cost of $(4n + 8p)$ flops per rotation use to the OPAST's complexity. The global complexity remains equal to $3np + p^2 + O(p + n)$.

4. Data whitening

We propose here to exploit the GOPAST to derive a fast adaptive whitening algorithm. Several applications consider the following instantaneous mixture model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \tag{11}$$

where $\mathbf{x}(t) = [x_1(t) \cdots x_n(t)]^T$ is the observation vector, $\mathbf{A}$ is an $n \times p$ full column rank mixture matrix, $\mathbf{s}(t) = [s_1(t) \cdots s_p(t)]^T$ is the vector of statistically independent sources and $\mathbf{n}(t)$ is the observed noise of covariance matrix $\sigma^2\mathbf{I}_n$.

We seek to transform the mixture matrix $\mathbf{A}$ into a unitary matrix while projecting the observed vector $\mathbf{x}(t)$ onto the principal subspace spanned by the column vectors of $\mathbf{A}$. The used matrix in this transformation is called the whitening matrix and denoted $\mathbf{S}$.

If noise is negligible, it is shown in [33] that $\mathbf{S}$ can be obtained from the eigendecomposition of $\mathbf{C}_{xx}$ according to

$$\mathbf{S} = \mathbf{\Lambda}_p^{-\frac{1}{2}}\mathbf{V}_p^H \tag{12}$$

where $\mathbf{V}_p$ is the $n \times p$ matrix of the principal eigenvectors of $\mathbf{C}_{xx}$ and $\mathbf{\Lambda}_p$ is the diagonal matrix of their associated principal eigenvalues.

In our adaptive scheme, an estimate of $\mathbf{V}_p$ at time t is given by the weight matrix $\mathbf{W}(t)$ while an estimate of $\mathbf{\Lambda}_p^{-1}$ is given by $\operatorname{diag}(\mathbf{Z}(t))$, i.e. the diagonal matrix formed by the diagonal entries of $\mathbf{Z}(t)$. Hence an estimate of whitening matrix $\mathbf{S}$ at time t can be obtained as:

$$\mathbf{S}(t) = \left[\operatorname{diag}\left(\mathbf{Z}(t)\right)\right]^{\frac{1}{2}}\mathbf{W}^H(t) \tag{13}$$

where $[\operatorname{diag}(\mathbf{Z}(t))]^{\frac{1}{2}}$ denotes the square root of the diagonal matrix $\operatorname{diag}(\mathbf{Z}(t))$.

Now, if noise is non-negligible (i.e., σ^2 is large), it is shown in [34] that the whitening matrix becomes:

$$\mathbf{S} = \left[\mathbf{\Lambda}_p - \sigma^2\mathbf{I}_p\right]^{-\frac{1}{2}}\mathbf{V}_p^H \tag{14}$$

In this case, the adaptive estimation of $\mathbf{S}$ requires the a priori estimation of the noise power σ^2. The latter can be achieved by observing that:

$$\begin{cases} \operatorname{trace}(\mathbf{C}_{xx}) = \operatorname{trace}(\mathbf{\Lambda}_p) + (n-p)\sigma^2 \\ \operatorname{trace}\left(\left[\operatorname{diag}(\mathbf{Z})\right]^{-1}\right) = \operatorname{trace}(\mathbf{\Lambda}_p) \end{cases} \tag{15}$$

which leads to

$$\sigma^2 = \frac{\operatorname{trace}(\mathbf{C}_{xx}) - \operatorname{trace}([\operatorname{diag}(\mathbf{Z})]^{-1})}{n-p} \tag{16}$$

In the adaptive scheme, we estimate the numerator's terms as

$$\operatorname{trace}\left(\mathbf{C}_{xx}(t)\right) = \beta\operatorname{trace}\left(\mathbf{C}_{xx}(t-1)\right) + (1-\beta)\left\|\mathbf{x}(t)\right\|^2 \tag{17}$$

$$\operatorname{trace}\left(\left[\operatorname{diag}\left(\mathbf{Z}(t)\right)\right]^{-1}\right) = \sum_{i=1}^{p}\frac{1}{\mathbf{Z}_{ii}(t)} \tag{18}$$

Finally, the estimate of the whitening matrix at time t is given by

$$\mathbf{S}(t) = \left[\left[\operatorname{diag}\left(\mathbf{Z}(t)\right)\right]^{-1} - \hat{\sigma}^2(t)\mathbf{I}_p\right]^{-\frac{1}{2}}\mathbf{W}^H(t) \tag{19}$$

where $\hat{\sigma}^2(t)$ refers to the estimate of the noise power given by (16)–(18). Notation $()^{-\frac{1}{2}}$ refers to the inverse square root of the diagonal matrix into consideration.

5. MSA and MCA algorithms

Minor components or subspace analysis (MCA or MSA) is known to be a more difficult problem than PCA or PSA. For this reason, we propose to transform the former problem into a PCA problem that can be solved using one of the proposed algorithms.

Different solutions are considered next and compared later by simulation experiments:

- Complete EVD method: All eigenvectors of $\mathbf{C}_{xx}$ are estimated and tracked by a Jacobi-like algorithm.
- α-GOPAST methods: Two methods are considered here using H-GOPAST to extract the principal components of the transformed covariance matrix $\mathbf{C}'_{xx} = \alpha\mathbf{I} - \mathbf{C}_{xx}$ which correspond to the minor components of $\mathbf{C}_{xx}$. This transformation produces a slight modification in H-GOPAST equations as shown next.

5.1. Complete EVD method

It consists of diagonalizing the data covariance matrix $\mathbf{C}_{xx}$ using Jacobi-like algorithm in an adaptive scheme. This method extracts all eigenvectors and eigenvalues which costs $O(n^2)$ flops as determined below. The covariance matrix being Hermitian, its eigendecomposition can be written as

$$\mathbf{C}_{xx} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H \tag{20}$$

where $\mathbf{V}$ is unitary and $\mathbf{\Lambda}$ is diagonal. $\mathbf{V}$ can be estimated as the unitary matrix that diagonalizes $\mathbf{C}_{xx}$ using a product of Givens rotations as explained in Section 3.2. In the adaptive scheme, the update of the diagonalized matrix $\mathbf{\Lambda}(t)$ is computed as

$$\mathbf{\Lambda}(t) = \beta\mathbf{\Lambda}(t-1) + (1-\beta)\mathbf{y}(t)\mathbf{y}^H(t) \tag{21}$$

where $\mathbf{y}(t) = \mathbf{V}^H(t-1)\mathbf{x}(t)$.

Here, we propose to use only two Givens rotations at each time instant t where the Givens rotation indices using Hybrid criterion and the rotation angles are computed as shown in Sections 3.2.1 and 3.2.2, respectively. Note that using one pair of Givens rotations is sufficient to have a good performance as shown in the simulation results (cf. Fig. 4).

This method is of moderate computational cost and is considered only for the comparison with our proposed low complexity methods. Its global cost is equal to $2n^2 + 12n + O(1)$ flops per time instant (if many iterations are considered per time instant, the global cost increases by $12n$ flops per additional iteration).

5.2. α-GOPAST methods

The principle of these methods consists of transforming the data covariance matrix in such a way the desired minor eigenvectors can be extracted as the principal eigenvectors of the transformed matrix. Based on the eigendecomposition of $\mathbf{C}_{xx}$ given in (20), one can write

$$\mathbf{C}'_{xx} = \alpha\mathbf{I} - \mathbf{C}_{xx} = \mathbf{V}[\alpha\mathbf{I} - \mathbf{\Lambda}]\mathbf{V}^H \tag{22}$$

where α is a scalar chosen to be larger than the largest eigenvalue of $\mathbf{C}_{xx}$.

$\mathbf{C}'_{xx}$ and $\mathbf{C}_{xx}$ have the same eigenvectors basis except that, by this transformation, the eigenvectors of $\mathbf{C}'_{xx}$ are associated to the eigenvalues $\lambda'_i = \alpha - \lambda_i$, $i = 1, \dots, n$, where λ_i is the ith eigenvalue of $\mathbf{C}_{xx}$. One can easily observe that the minor eigenvectors of $\mathbf{C}_{xx}$ are now associated to the largest eigenvalues of $\mathbf{C}'_{xx}$ and therefore the MCA of $\mathbf{C}_{xx}$ becomes PCA of $\mathbf{C}'_{xx}$. Here, we propose two versions of this method:

(1) *α-GOPAST1*: It consists of generating a random vector satisfying $\mathbf{C}_{bb} = \mathbf{E}[\mathbf{b}(t)\mathbf{b}^H(t)] = \alpha\mathbf{I}$, then the transformed covariance matrix $\mathbf{C}'_{xx}$ is evaluated in the adaptive scheme as

$$\mathbf{C}'_{xx}(t) = \beta\mathbf{C}'_{xx}(t-1) + \mathbf{b}(t)\mathbf{b}^H(t) - \mathbf{x}(t)\mathbf{x}^H(t) \tag{23}$$

Next, we use the same approach as H-GOPAST algorithm using the updated covariance matrix in (23) instead of (1). By replacing (23) in (3), we obtain:

$$\begin{aligned}\mathbf{W}(t) = {} & \left[\beta\mathbf{C}'_{xx}(t-1)\mathbf{W}(t-1) + \mathbf{XJY}^H\right] \\ & \times\left[\mathbf{W}(t-1)^H\mathbf{C}'_{xx}(t-1)\mathbf{W}(t-1) + \mathbf{YJY}^H\right]^{-1}\end{aligned} \tag{24}$$

where $\mathbf{X} = [\mathbf{b}(t)\ \mathbf{x}(t)]$, $\mathbf{Y} = \mathbf{W}^H(t-1)\mathbf{X}$ and $\mathbf{J} = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$. Using projection approximation and Schur's inversion lemma [35] leads to

$$\begin{aligned}\mathbf{W}(t) &= \mathbf{W}(t-1) - \frac{1}{\beta}\mathbf{W}(t-1)\mathbf{Y\Lambda Q}^H + \frac{1}{\beta}\mathbf{XJQ}^H \\ &\quad - \frac{1}{\beta^2}\mathbf{XJQ}^H\mathbf{Y\Lambda Q}^H \\ &= \mathbf{W}(t-1) + \mathbf{PQ}^H\end{aligned} \tag{25}$$

where $\mathbf{P} = -\frac{1}{\beta}\mathbf{W}(t-1)\mathbf{Y\Lambda} + \frac{1}{\beta}\mathbf{XJ} - \frac{1}{\beta^2}\mathbf{XJQ}^H\mathbf{Y\Lambda}$, $\mathbf{Q} = \mathbf{Z}(t-1)\mathbf{Y}$, and $\mathbf{\Lambda} = (\mathbf{J} + \frac{1}{\beta}\mathbf{Y}^H\mathbf{Z}(t-1)\mathbf{Y})^{-1}$ which is the inverse of a 2 × 2 matrix. $\mathbf{Z}(t)$ refers to $(\mathbf{W}^H(t-1)\mathbf{C}'_{xx}(t)\mathbf{W}(t-1))^{-1}$ and can be updated with linear complexity as follows

$$\mathbf{Z}(t) = \frac{1}{\beta}\mathbf{Z}(t-1) - \frac{1}{\beta^2}\mathbf{Q\Lambda Q}^H \tag{26}$$

To get an orthonormal basis, we use the following fast orthonormalization technique:

$$\mathbf{W}^{\perp}(t) = \mathbf{W}(t)\left(\mathbf{W}^H(t)\mathbf{W}(t)\right)^{-\frac{1}{2}} \tag{27}$$

Taking into account the orthonormality of $\mathbf{W}(t-1)$, a straightforward calculation leads to:

$$\mathbf{W}^{\perp}(t) = \mathbf{W}(t)\left(\mathbf{I} + \mathbf{M}(t)\right)^{-\frac{1}{2}} \tag{28}$$

where $\mathbf{M}(t) = \mathbf{QP}^H\mathbf{PQ}^H$ and $(\mathbf{I} + \mathbf{M}(t))^{-\frac{1}{2}}$ stands for the inverse square root of $\mathbf{I} + \mathbf{M}(t)$ which can be computed with a linear complexity according to the following proposition.

Proposition 5.1. *Assume that* $\mathbf{M} = \mathbf{EDE}$ *with* $\mathbf{E}$ *orthonormal,* $\mathbf{D} = \operatorname{diag}(\alpha_1, \dots, \alpha_d)$ *and* $\mathbf{D} + \mathbf{I}$ *positive definite. Then*:

$$\begin{aligned}&\left(\mathbf{I} + \mathbf{M}(t)\right)^{-\frac{1}{2}} = \mathbf{I} + \mathbf{ED}'\mathbf{E} \\ &\quad \textit{where } \mathbf{D}' = \operatorname{diag}\left(\frac{1}{1+\alpha_1} - 1, \dots, \frac{1}{1+\alpha_d} - 1\right)\end{aligned} \tag{29}$$

In our case, $\mathbf{M}$ is of rank 2 (i.e., $d = 2$) and its columns space is spanned by the 2 column vectors of $\mathbf{Q}$. To decompose $\mathbf{M}$ into the form given in the above proposition, let us write $\mathbf{E} = \mathbf{QV}$ where $\mathbf{V}$

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

Table 3
α-GOPAST1.

```
Initialization: $\mathbf{W}(0) = \left[\begin{smallmatrix}\mathbf{I}_p\\ \mathbf{0}\end{smallmatrix}\right]$, $\mathbf{Z}(0) = \mathbf{I}_p$ and $\mathbf{J} = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$
Adaptation:
  Generate random vector $\mathbf{b}(t)$ with $\mathbf{C}_{bb} = \alpha\mathbf{I}$
  $\mathbf{X} = [\mathbf{b}(t)\ \mathbf{x}(t)]$
  $\mathbf{Y} = \mathbf{W}^H\mathbf{X}$
  $\mathbf{Q} = \mathbf{Z}(t-1)\mathbf{Y};\ \boldsymbol{\Lambda} = (\mathbf{J} + \frac{1}{\beta}\mathbf{Y}^H\mathbf{Q});$
  $\mathbf{P} = -\frac{1}{\beta}\mathbf{W}\mathbf{Y}\boldsymbol{\Lambda} + \frac{1}{\beta}\mathbf{X}\mathbf{J} - \frac{1}{\beta^2}\mathbf{X}\mathbf{J}\mathbf{Q}^H\mathbf{Y}\boldsymbol{\Lambda};$
  $\mathbf{M} = \mathbf{Q}(\mathbf{P}^H\mathbf{P})\mathbf{Q}^H;$
  $\mathbf{R} = \mathrm{pinv}(\mathbf{Q})\mathbf{M}\mathbf{Q};$
  $[\mathbf{V}\ \mathbf{D}] = \mathrm{eig}(\mathbf{R})$
  $\mathbf{E} = \mathbf{Q}\mathbf{V}$
  Normalize the columns of $\mathbf{E}$
  $\mathbf{D}' = \mathrm{diag}(\frac{1}{\sqrt{1+D(1)}-1}, \frac{1}{\sqrt{1+D(2)}-1})$
  $\mathbf{P}' = \mathbf{P} + \mathbf{W}(t-1)\mathbf{E}\mathbf{D}'\mathbf{V} + \mathbf{P}\mathbf{Q}^H\mathbf{E}\mathbf{D}'\mathbf{V}$
  $\mathbf{W} = \mathbf{W} + \mathbf{P}'\mathbf{Q}^H$
  $\mathbf{Z}(t) = \frac{1}{\beta}\mathbf{Z}(t-1) - \frac{1}{\beta^2}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^H$
  Givens rotations: Table 2.
```

Table 4
α-GOPAST2.

```
Initialization: $\mathbf{W}(0) = \left[\begin{smallmatrix}\mathbf{I}_p\\ \mathbf{0}\end{smallmatrix}\right]$, $\mathbf{Z}(0) = \mathbf{I}_p$
Adaptation:
  $\mathbf{y}(t) = \mathbf{W}^H(t-1)\mathbf{x}(t)$
  $\alpha(t) = \beta\alpha(t) + (1-\beta)\|\mathbf{x}(t)\|^2$
  $\mu = -\frac{1-\beta}{\beta+\alpha(t)(1-\beta)}$
  $\mathbf{z}(t) = \mathbf{W}(t-1)\mathbf{y}(t)$
  $\mathbf{v}(t) = \frac{\mathbf{z}(t)}{\|\mathbf{y}(t)\|} + \mu\|\mathbf{y}(t)\|\mathbf{x}(t)$
  $\mathbf{q}(t) = \frac{\mathbf{v}(t)}{\|\mathbf{v}(t)\|} - \frac{\mathbf{z}(t)}{\|\mathbf{y}(t)\|}$
  $\mathbf{W}(t) = \mathbf{W}(t-1) + \frac{1}{\|\mathbf{y}(t)\|}\mathbf{q}(t)\mathbf{y}^H(t)$
  $\mathbf{C}_{yy}(t) = \beta\mathbf{C}_{yy}(t-1) + (1-\beta)\mathbf{y}(t)\mathbf{y}^H(t)$
  Givens rotations: Table 2.
```

is a 2×2 unknown matrix we need to estimate. For that, we can observe that

$$\mathbf{R} = \mathbf{Q}^{\#}\mathbf{M}\mathbf{Q} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \tag{30}$$

where $\mathbf{Q}^{\#}$ denotes the pseudo-inverse of $\mathbf{Q}$.

In other words, $\mathbf{V}$ and $\mathbf{D}$ can be computed[6] via the eigendecomposition of the 2×2 matrix $\mathbf{R}$. Finally, by applying (29) to (28), we obtain

$$\mathbf{W}^{\perp}(t) = \mathbf{W}(t-1) + \mathbf{P}'\mathbf{Q}^H \tag{31}$$

where $\mathbf{P}' = \mathbf{P} + \mathbf{W}(t-1)\mathbf{E}\mathbf{D}'\mathbf{V} + \mathbf{P}\mathbf{Q}^H\mathbf{E}\mathbf{D}'\mathbf{V}$.

This algorithm's version is summarized in Table 3.

(2) *α-GOPAST2*: It is based on power method followed by orthonormalization step, i.e.

$$\begin{aligned}\mathbf{W}_+(t) &= \mathbf{C}'_{xx}(t)\mathbf{W}(t-1)\\ \mathbf{W}(t) &= \mathrm{orth}\big(\mathbf{W}_+(t)\big)\end{aligned} \tag{32}$$

where the transformed covariance matrix $\mathbf{C}'_{xx}(t)$ is evaluated in the adaptive scheme using Eq. (1) as follows

$$\begin{aligned}\mathbf{C}'_{xx}(t) &= \alpha\mathbf{I} - \mathbf{C}_{xx}(t)\\ &= \beta\big(\alpha\mathbf{I} - \mathbf{C}_{xx}(t-1)\big) + (1-\beta)\big(\alpha\mathbf{I} - \mathbf{x}(t)\mathbf{x}^H(t)\big)\\ &= \beta\mathbf{C}'_{xx}(t-1) + \alpha(1-\beta)\mathbf{I} - (1-\beta)\mathbf{x}(t)\mathbf{x}^H(t)\end{aligned} \tag{33}$$

Replacing (33) in (32) and assuming a slow variation of the weight matrix that allows us to approximate at the first order $\mathbf{C}_{xx}(t-1)\mathbf{W}(t-1)$ by $\mathbf{C}_{xx}(t-1)\mathbf{W}(t-2)$ (this is known as projection approximation [12]) and to assume the quasi-orthogonality of the updated weight matrix, yields

$$\begin{aligned}\mathbf{W}_+(t) &= \mathbf{C}'_{xx}(t)\mathbf{W}(t-1)\\ &= \gamma\left\{\mathbf{W}(t-1) - \frac{(1-\beta)}{\gamma}\mathbf{x}(t)\mathbf{y}^H(t)\right\}\end{aligned} \tag{34}$$

where $\gamma = \beta + \alpha(1-\beta)$ and $\mathbf{y}(t) = \mathbf{W}^H(t-1)\mathbf{x}(t)$. One can observe that our updated equation (34) is very similar to the proposed one in DPM algorithm [36] except that the latter is based on the gradient method in which the choice of the step of adaptation is not obvious. Our method is based on projection approximation and the step of adaptation ($-\frac{1-\beta}{\gamma}$) comes in a natural way and it depends on the forgetting factor β and the scalar α (which is computed in the adaptive scheme as the estimate of $\mathrm{trace}(\mathbf{C}_{xx}(t))$ given by $\alpha(t) = \beta\alpha(t-1) + (1-\beta)\|\mathbf{x}(t)\|^2$).

In the literature, one can find different ways to perform the orthonormalization step [18,37–39]. The most stable of them are presented in [38,18]. The technique used in the FDPM algorithm [38] costs $6np$ flops per iteration while the proposed algorithm FSDPM (Fast and Stable Data Projection Method) in [18] costs $3np$ flops per iteration. Hence, we propose to use the same orthonormalization technique as the one presented in [18] which is the less expensive one that preserves the algorithm's stability.

Regarding the extraction of the eigenvectors, we diagonalize the $p \times p$ covariance matrix $\mathbf{C}_{yy} = E[\mathbf{y}\mathbf{y}^H] = \mathbf{W}^H\mathbf{C}_{xx}\mathbf{W}$ using Givens rotations. The latter covariance matrix is updated adaptively as $\mathbf{C}_{yy}(t) = \beta\mathbf{C}_{yy}(t-1) + (1-\beta)\mathbf{y}(t)\mathbf{y}^H(t)$.

This algorithm's version is summarized in Table 4.

5.3. OCA algorithm for minor subspace

In the case where we are interested in the MSA only, introduce here the Orthogonal Complement Algorithm (OCA) which computes the desired minor subspace as the orthogonal complement of the principal subspace computed by OPAST algorithm. The implementation details of this algorithm are given in Appendix A and the corresponding simulation results are discussed in Section 6.

6. Numerical results and discussion

In the present section, we show the performance of the new versions of GOAPST algorithm which are compared in the context of PCA with SVD algorithm. The performance of MSA and MCA algorithms are compared, for the benchmark, with YAST [24] and MC-YAST [40], respectively. They are also compared with an existing linear complexity algorithm FSDPM [18] which has the advantage of proved stability as compared to other existing MCA methods, e.g. [41]. The used performance factors are given in the following:

- Subspace performance

$$\zeta(t) = \frac{1}{r}\sum_{i=1}^{r}\frac{\mathrm{trace}(\mathbf{W}_i^H(t)(\mathbf{I} - \mathbf{W}_{ex}\mathbf{W}_{ex}^H)\mathbf{W}_i(t))}{\mathrm{trace}(\mathbf{W}_i^H(t)(\mathbf{W}_{ex}\mathbf{W}_{ex}^H)\mathbf{W}_i(t))}$$

- Eigenvectors performance

$$\rho(t) = \frac{1}{rp}\sum_{i=1}^{r}\big\|\mathbf{W}_i(t) - \mathbf{W}_{ex}\big\|^2$$

[6] The eigenvectors being computed up to a scalar constant, we need to normalize the columns of $\mathbf{E}$ as shown in Table 3.

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

where $r = 150$ is the number of Monte Carlo runs, $\mathbf{W}_i(t)$ is the matrix of desired eigenvectors (resp. subspace) at experience i and iteration t, $\mathbf{W}_{ex}$ is the exact matrix of eigenvectors (resp. subspace) computed from the exact data covariance matrix. Note that, since the unit-norm eigenvectors are estimated up to unknown phase factors, we normalized them in such a way their first entries are non-negative real-valued in order to compute the least squares error criterion $\rho(t)$.

Regarding the whitening algorithm, we compare it with the RLS-type algorithm in [42]. The whitening quality is evaluated by using the following performance factor

$$\eta(t) = \frac{1}{r}\sum_{i=1}^{r}\left\|\mathbf{S}_i(t)\mathbf{C}_{xx}(t)\mathbf{S}_i^H(t) - \mathbf{I}_r\right\|^2 \qquad (35)$$

where $\mathbf{S}_i(t)$ is the whitening matrix at the ith run and tth iteration.

The data is generated in the following way:

- For the PCA, we generate data according to the model

  $$\mathbf{x}(t) = \mathbf{A}(t)\mathbf{s}(t) + \mathbf{n}(t) \qquad (36)$$

  where the $n \times p$ matrix $\mathbf{A}$ corresponds to the steering matrix on a uniform linear array [43] (the source angles of arrival are generated randomly) and $\mathbf{s}(t)$ (resp. $\mathbf{n}(t)$) is a $p \times 1$ unit norm Gaussian random source signal (resp. an $n \times 1$ white noise signal of covariance $\sigma^2\mathbf{I}$).
- For MCA or MSA, the data is generated according to the model[7]

  $$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \qquad (37)$$

  where $\mathbf{A}$ is the Hermitian $n \times n$ square root of the 'exact' covariance matrix $\mathbf{R} = \mathbf{W}\mathbf{D}\mathbf{W}^H$ where $\mathbf{D}$ is a positive diagonal matrix and $\mathbf{W}$ is an orthonormal matrix randomly selected. The $n \times 1$ source vector $\mathbf{s}(t)$ is Gaussian and white, i.e., $\mathbf{E}[\mathbf{s}(t)\mathbf{s}^H(t)] = \mathbf{I}$.

6.1. GOPAST versions comparison

For small system dimensions (Fig. 1), one can observe that all GOPAST's versions have similar performance in terms of estimation accuracy and convergence rate. This observation can be justified by the fact that the desired signal dimension is small (i.e., $p = 3$) and hence the selection of the rotation indices for the diagonalization of the 3×3 matrix $\mathbf{Z}$ has little impact on the algorithm's performance. We observe also that the proposed algorithms have the same performance as the SVD applied to the sample covariance matrix (1).

For Figs. 2 and 3, we have chosen a larger subspace rank, i.e., $p = 9$ (resp. $p = 40$) where the difference in the algorithms performance appears. In this case, $\mathbf{Z}$ becomes bigger and there is a great difference between its off-diagonal elements. One can observe that IMED-GOPAST outperforms MED-GOPAST due to the improved selection criterion of rotation indices. AS-GOPAST has a good estimation accuracy but it suffers from low speed of convergence. H-GOPAST presents a good trade-off between IMED-GOPAST and AS-GOPAST, especially when the principal subspace rank is large.

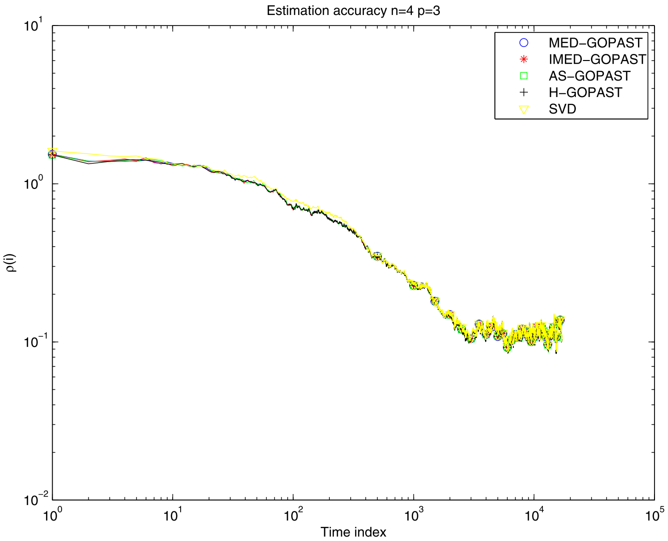

Fig. 1. Principal eigenvectors estimation ($n = 4$, $p = 3$).

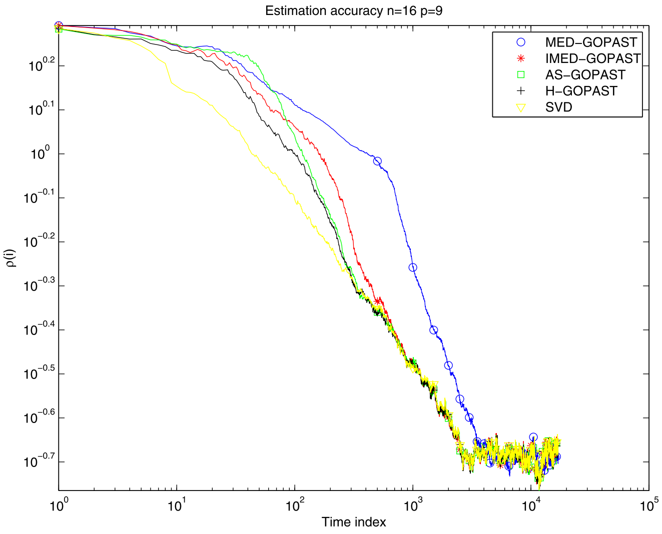

Fig. 2. Principal eigenvectors estimation ($n = 16$, $p = 9$).

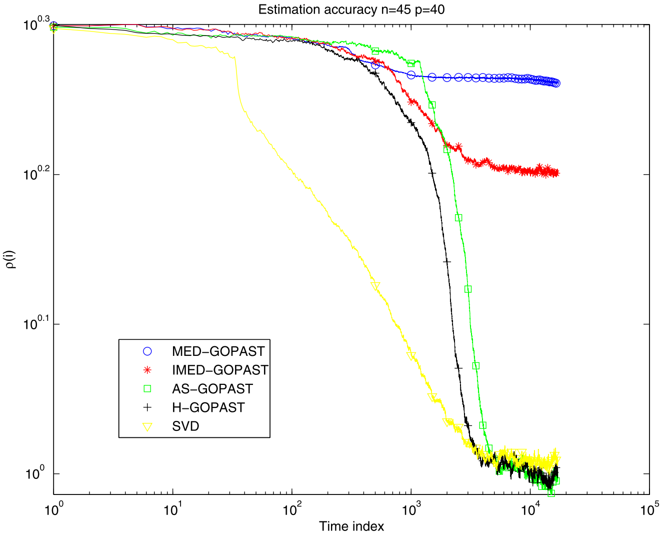

Fig. 3. Principal eigenvectors estimation ($n = 45$, $p = 40$).

[7] This model is chosen in such a way the minor eigenvalues are distinct two by two (which is not the case for the model in (36) where all minor eigenvalues are equal to σ^2).

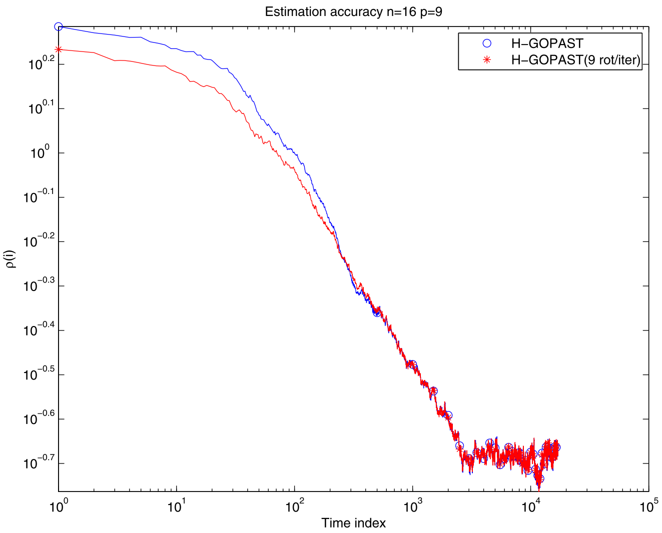

Fig. 4. Multiple Givens rotations effect.

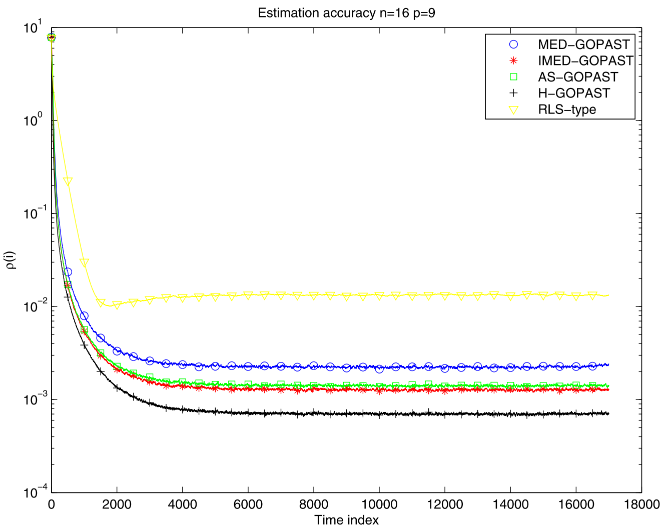

Fig. 5. Whitening performance ($n = 16$, $p = 9$).

Fig. 4 shows that performing multiple rotations per time instant does not increase the steady state performance but it improves slightly the convergence rate.

6.2. Data whitening comparison

The four adaptive whitening algorithms outperform the RLS-type [42] whitening algorithm in terms of accuracy as shown in Fig. 5. Again, we observe that the H-GOPAST version is the best. Note that, for this simulation, we have considered a large SNR value and hence Eq. (13) is used to compute the whitening matrix $\mathbf{S}$.

In many applications (e.g. source separation problem), one needs to whiten the noise-free covariance matrix, i.e. matrix $\mathbf{C}_{xx} - \sigma^2\mathbf{I}$. As we mentioned earlier, if the noise power is not negligible, we need to use Eq. (19) for the updating of the whitening matrix and hence we have to replace in criterion (35) matrix $\mathbf{C}_{xx}$ by its noise-free counterpart $\mathbf{C}_{xx} - \sigma^2\mathbf{I}$.

Fig. 6 illustrates the gain we obtain in this case when considering the corrected algorithm's version (red curve) of Eq. (19) as compared to the non-corrected one (blue curve) of Eq. (13).

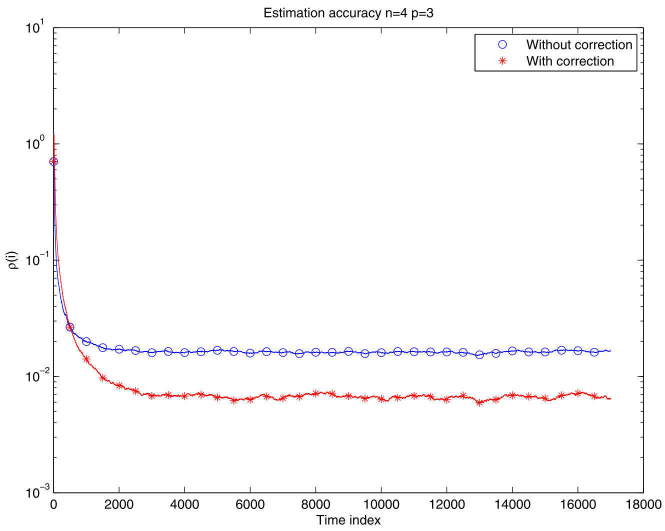

Fig. 6. Proposed correction for whitening algorithm in low SNR (0 dB). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

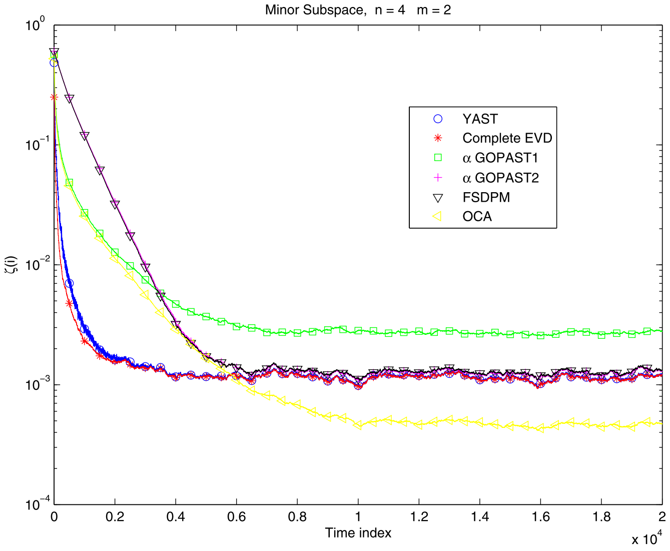

Fig. 7. Minor subspace estimation ($n = 4$, $p = 2$). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

6.3. MSA and MCA comparison

For small system dimensions, the proposed algorithms show a good performance to estimate the minor subspace (Fig. 7) and the minor components (Fig. 8) of the observed data covariance matrix. As a benchmark, the performance results of YAST (blue curve) and Complete EVD (green curve) are also provided. We can observe that OCA has a good estimation accuracy but takes more time to converge as compared to α-GOPAST1 and α-GOPAST2. Also, α-GOPAST2 and FSDPM [18] have almost the same performance.

Many observations can be done from these figures:

- α-GOPAST1 has faster convergence rate as compared to α-GOPAST2 and OCA but it suffers from poor estimation accuracy (especially when p is large as shown in Figs. 9 and 10).
- For MSA, OCA has better estimation accuracy than the other algorithms but suffers from low convergence rate.

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

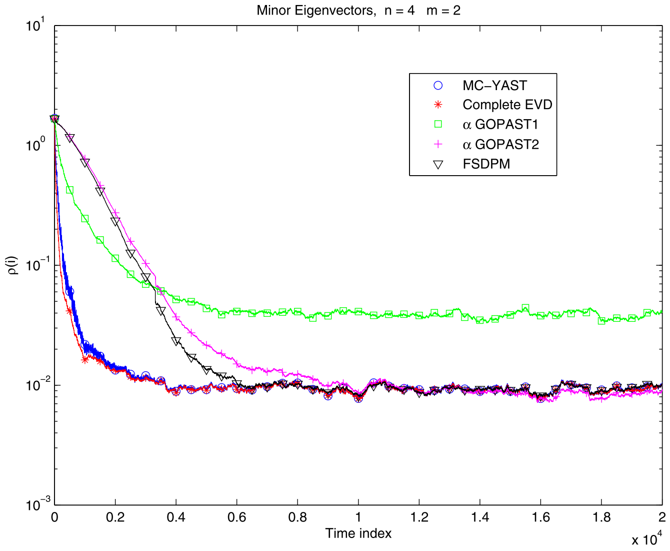

Fig. 8. Minor eigenvectors estimation ($n = 4$, $p = 2$). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

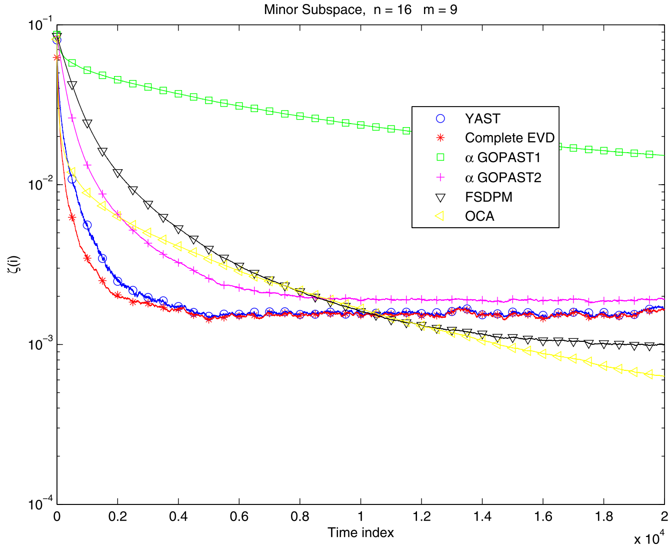

Fig. 9. Minor subspace estimation ($n = 16$, $p = 9$).

- For the MCA, α-GOPAST2 has similar performance as FSDPM but with the advantage of an ‘automatic’ selection of the algorithm’s step.[8]

In Figs. 9 and 10, the minor subspace rank is increased to $p = 9$ in which case the previous observations are confirmed and the poor performance estimation of α-GOPAST1 is highlighted.

7. Conclusion

In this paper, we proposed new algorithms for the extraction and tracking of principal and minor eigenvectors of a given data covariance matrix. For the PCA, we have improved the performance of GOAPST algorithm in terms of convergence rate and estimation accuracy thanks to an appropriate selection procedure of the Givens rotation indices. Then, we exploited the previous PCA algorithms to derive a very efficient data whitening method that outperforms the existing methods of similar complexity. Finally, we have proposed new algorithms with linear complexity to solve the MSA or MCA problems. The proposed method has some advantages in terms of complexity, convergence rate and estimation accuracy.

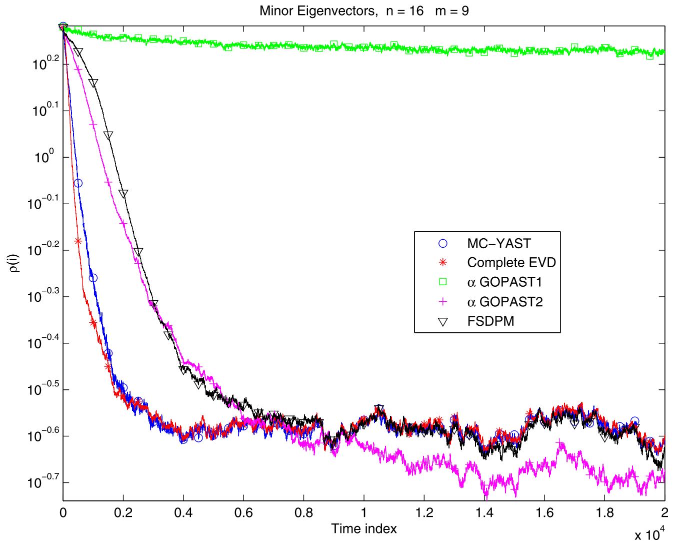

Fig. 10. Minor eigenvectors estimation ($n = 16$, $p = 9$).

Appendix A. OCA algorithm

A.1. Orthogonal Complement Algorithm (OCA) for MSA

Taking advantages of linear complexity and the efficiency of OPAST algorithm to estimate and track the principal subspace, we propose here an adaptive algorithm for minor subspace tracking based on the fast computation of the orthogonal complement of the principal subspace produced by OPAST.

Assume that $\mathbf{W}(t) \in \mathbb{C}^{n\times m}$, where $m = n - p$, is the weight matrix of principal subspace and $\mathbf{U}(t) \in \mathbb{C}^{n\times p}$ is the complementary subspace orthogonal to $\mathbf{W}(t)$.

Breaking down $\mathbf{W}(t)$ as

$$\mathbf{W}(t) = \begin{bmatrix} \mathbf{W}_1(t) \\ \mathbf{W}_2(t) \end{bmatrix}$$

where $\mathbf{W}_1(t) \in \mathbb{C}^{p\times m}$ and $\mathbf{W}_2(t) \in \mathbb{C}^{m\times m}$ and considering that the m-rows of $\mathbf{W}_2(t)$ are linearly independent, allows us to write the rows of $\mathbf{W}_1(t)$ in terms of those of $\mathbf{W}_2(t)$ as follows

$$\mathbf{W}_1(t) = \mathbf{P}^H(t)\mathbf{W}_2(t) \tag{38}$$

Hence, the transition matrix $\mathbf{P}(t) \in \mathbb{C}^{m\times p}$ can be calculated as

$$\mathbf{P}^H(t) = \mathbf{W}_1(t)\mathbf{W}_2^{-1}(t) \tag{39}$$

One can see that

$$\mathbf{U}(t) = \begin{bmatrix} \mathbf{I}_p \\ -\mathbf{P}(t) \end{bmatrix} \tag{40}$$

is orthogonal to $\mathbf{W}(t)$ (i.e., $\mathbf{U}^H(t)\mathbf{W}(t) = \mathbf{W}_1(t) - \mathbf{P}^H(t)\mathbf{W}_2(t) = \mathbf{0}$) and consequently its column vectors form a basis of the desired minor subspace.

The OCA computes the current subspace weight matrix $\mathbf{U}(t)$ by successively computing the following:

(1) an orthonormal basis $\mathbf{W}(t)$ using OPAST algorithm;
(2) the inverse of $\mathbf{W}_2(t)$ adaptively;

[8] Note, that for these numerical experiments, we have chosen (by trials) the best step value for FSDPM. Indeed, a poor step value selection leads to poor convergence performance or even to the algorithm’s divergence.

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

Table 5
Orthogonal complement algorithm.

```
Initialization: $\mathbf{W}(0) =$ orthonormal basis, $\mathbf{Z}(0) = \mathbf{I}_p$,
  $\mathbf{P}^H(0) = \mathbf{W}_1(0)\mathbf{W}_2^{-1}(0)$
Adaptation: For $t \geqslant 1$
  Execute OPAST (Table 1)
  $\mathbf{w}_e = \mathbf{W}_2^{-1}(t-1)\mathbf{e}_2$
  $\mathbf{w}_q = \mathbf{W}_2^{-H}(t-1)\mathbf{q}$
  $f = -\frac{1}{1+\mathbf{q}^H\mathbf{w}_e}$
  $\mathbf{h} = \mathbf{e}_1 + f\mathbf{P}^H(t-1)\mathbf{e}_2 + f\mathbf{e}_1(\mathbf{w}_q^H\mathbf{e}_2)$
  $\mathbf{w}_p = \mathbf{P}^H(t-1)\mathbf{w}_q$
  $\mathbf{P}(t) = \mathbf{P}(t-1) + \mathbf{w}_q\mathbf{h}^H$
  $\mathbf{U}(t) = \left[\begin{smallmatrix}\mathbf{I}_p \\ -\mathbf{P}(t)\end{smallmatrix}\right]$
```

(3) a transition matrix $\mathbf{P}(t)$.

Fast updating of $\mathbf{W}_2^{-1}(\mathbf{t})$: From the last equation of Table 1, $\mathbf{W}_1(t)$ and $\mathbf{W}_2(t)$ are formed as

$$\mathbf{W}_1(t) = \mathbf{W}_1(t-1) + \mathbf{e}_1\mathbf{q} \tag{41}$$

$$\mathbf{W}_2(t) = \mathbf{W}_2(t-1) + \mathbf{e}_2\mathbf{q} \tag{42}$$

where the vector $\mathbf{e}_1$ (resp. $\mathbf{e}_2$) contains the first p elements (resp. last m elements) of vector $\mathbf{e}$. $\mathbf{W}_2^{-1}(t)$ is updated adaptively by using Schur's lemma [35] as

$$\mathbf{W}_2^{-1}(t) = \mathbf{W}_2^{-1}(t-1) + f\mathbf{w}_e\mathbf{w}_q^H \tag{43}$$

where $f = -\frac{1}{1+\mathbf{q}^H\mathbf{w}_e}$, $\mathbf{w}_e = \mathbf{W}_2^{-1}(t-1)\mathbf{e}_2$ and $\mathbf{w}_q^H = \mathbf{q}^H\mathbf{W}_2^{-1}(t-1)$. This computation required $3m^2 + m$ flops per iteration.

Computation of the transition matrix $\mathbf{P}(\mathbf{t})$: Replacing (38) and (43) in (39), we obtain

$$\mathbf{P}^H(t) = \mathbf{P}^H(t-1) + \mathbf{h}\mathbf{w}_q^H \tag{44}$$

with $\mathbf{h} = \mathbf{e}_1 + f\mathbf{P}^H(t-1)\mathbf{e}_2 + f\mathbf{e}_1(\mathbf{w}_q^H\mathbf{e}_2)$. Updating $\mathbf{P}(t)$ required an additional cost equal to $2mp + m$ flops per iteration.

The weight matrix of minor subspace $\mathbf{U}(t)$ is obtained by replacing (44) in (40)

$$\mathbf{U}(t) = \begin{bmatrix} \mathbf{I}_p \\ -(\mathbf{P}(t-1) + \mathbf{w}_q\mathbf{h}^H) \end{bmatrix} = \mathbf{U}(t-1) + \tilde{\mathbf{w}}_q\mathbf{h}^H \tag{45}$$

where $\tilde{\mathbf{w}}_q = [\mathbf{0}_{1\times p}\ \mathbf{w}_q^T]^T$.

In addition to the complexity of OPAST algorithm which is equal to $3nm + m^2 + O(m)$, the computation of this weight matrix provides an additional cost equals $2mp + 3m^2 + O(m)$. One can observe that the required complexity to compute the weight matrix of the minor subspace is still linear and equals $5nm + 2m^2 + O(m)$.

This algorithm is useful when $n \gg m$ or in the other way the range of the minor subspace p is large. The proposed algorithm is summarized in Table 5.

References

[1] W. Jiao, S. Yang, Y. Chang, DOA estimation of multiple convolutively mixed sources based on principle component analysis, in: Lecture Notes in Comput. Sci., vol. 5863, 2009, pp. 340–348.
[2] V. Solo, High dimensional point process system identification: PCA and dynamic index models, in: 45th IEEE Conference on Decision and Control, 2006, pp. 829–833.
[3] Chao-Cheng Tu, B. Champagne, Subspace tracking of fast time-varying channels in precoded MIMO-OFDM systems, in: ICASSP, 2009, pp. 2565–2568.
[4] Xenofon Doukopoulos, Power techniques for blind channel estimation in wireless communication systems, PhD thesis, IRISA-INRIA/Université de Rennes I, 2004.
[5] A. Serita, K. Hattori, C. Yoshino, M. Hayakawa, N. Isezaki, Principal component analysis and singular spectrum analysis of ULF geomagnetic data associated with earthquakes, Nat. Hazards Earth Syst. Sci. 5 (2005) 685–689.
[6] H. Zhou, P. Xiao, Robust multiuser detection using Kalman filter and windowed projection approximation subspace tracking algorithm, IET Commun. 5 (2011) 495–504.
[7] Liping Sun, Guoan Bi, Liren Zhang, Orthonormal subspace tracking algorithm for space–time multiuser detection in multipath CDMA channels, IEEE Trans. Vehicular Technol. 56 (2007) 3838–3845.
[8] S. Dib, M. Barkat, M. Grimes, PAST and OPAST algorithms for STAP in monostatic airborne radar, in: International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 2011, pp. 177–181.
[9] J.P. Delmas, Subspace tracking for signal processing, in: Tulay Adali, Simon Haykin (Eds.), Adaptive Signal Processing: Next Generation Solutions, John Wiley and Sons, Inc., 2010, Chapter 4.
[10] Steve Bartelmaos, Adaptive subspace tracking and mobile localization in wireless networks, PhD thesis, Telecom-Paristech, 2008.
[11] I. Ali, D.N. Kim, T.T. Jeong, A new subspace tracking algorithm using approximation of Gram–Schmidt procedure, in: International Conference on Information and Multimedia Technology (ICIMT), 2009, pp. 244–248.
[12] B. Yang, Projection approximation subspace tracking, IEEE Trans. Signal Process. 43 (1995) 95–107.
[13] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Netw. 2 (1989) 459–473.
[14] M. Moonen, P. Van Dooren, J. Vandewalle, Updating singular value decompositions: A parallel implementations, in: Proc. SPIE Adv. Alg. Arch. Sig. Proc., 1989.
[15] S. Ouyang, Z. Bao, Fast principal component extraction by a weighted information criterion, IEEE Trans. Signal Process. (2002).
[16] C. Chatterjee, Z. Kang, V.P. Roychowdhury, Algorithms for accelerated convergence of adaptive PCA, IEEE Trans. Neural Netw. 11 (2000) 338–355.
[17] Y.N. Rao, J.C. Principe, A fast, on-line algorithm for PCA and its convergence characteristics, Neural Netw. Signal Process. 1 (2000) 299–307.
[18] Rong Wang, Minli Yao, Daoming Zhang, Hongxing Zou, A novel orthonormalization matrix based fast and stable DPM algorithm for principal and minor subspace tracking, IEEE Trans. Signal Process. 60 (2011) 466–472.
[19] D. Peng, Z. Yi, A modified Oja Xu MCA learning algorithm and its convergence analysis, IEEE Trans. Circuits Syst. II 54 (2007) 348–352.
[20] Mao Ye, Xu-Qian Fan, Xue Li, A class of self-stabilizing MCA learning algorithms, IEEE Trans. Neural Netw. 17 (2006) 1634–1638.
[21] W. Rong, Y. Minli, Z. Daoming, Z. Hongxing, Stable and orthonormal OJA algorithm with low complexity, IEEE Signal Process. Lett. 18 (2011) 211–214.
[22] P. Comon, G.H. Golub, Tracking a few extreme singular values and vectors in signal processing, Proc. IEEE 78 (1990) 1327–1343.
[23] H. Sakai, K. Shmizu, A new adaptive algorithm for minor component analysis, Signal Process. 71 (1998) 301–308.
[24] R. Badeau, G. Richard, B. David, YAST algorithm for minor subspace tracking, in: ICASSP, 2006.
[25] Mu Zhou, Alle-Jan van der Veen, Stable subspace tracking algorithm based on signed URV decomposition, in: ICASSP, 2011, pp. 2720–2723.
[26] M. Thameri, A. Kammoun, K. Abed-Meraim, A. Belouchrani, Fast principal component analysis and data whitening algorithms, in: WoSSPA, 2011, pp. 139–142.
[27] K. Abed-Meraim, A. Chkeif, Y. Hua, Fast orthonormal PAST algorithm, IEEE Signal Process. Lett. 7 (2000) 60–62.
[28] S. Bartelmaos, K. Abed-Meraim, Fast principal component extraction using Givens rotations, IEEE Signal Process. Lett. 15 (2008) 369–372.
[29] R.J. Piechocki, N. Canagarajah, J.P. McGeehan, Improving the direction-of-arrival resolution via double code filtering in WCDMA, in: 3G Mobile Communication Technologies, 2000, pp. 204–207.
[30] Pierre Comon, Christian Jutten, Handbook of Blind Source Separation, Academic Press, Inc., 2009.
[31] K. Abed-Meraim, A. Chkeif, Y. Hua, S. Attallah, On a class of orthonormal algorithms for principal and minor subspace tracking, J. VLSI Signal Process. 31 (2001) 57–70.
[32] G.H. Golub, C.F. Van Loan, Matrix Computations, John Hopkins University Press, 1996.
[33] Xiaolong Zhu, Xianda Zhang, Jimin Ye, Natural gradient-based recursive least-squares algorithm for adaptive blind source separation, Sci. China 47 (2004) 55–65.
[34] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics, IEEE Trans. Signal Process. 45 (1997) 434–444.
[35] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing, third edition, Cambridge University Press, 2007.
[36] J.-F. Yang, M. Kaveh, Adaptive eigensubspace algorithms for direction or frequency estimation and tracking, IEEE Trans. Acoust. Speech Signal Process. 36 (1988) 241–251.
[37] S. Attallah, The generalized Rayleigh's quotient adaptive noise subspace algorithm: A householder transformation-based implementation, IEEE Trans. Circuits Syst. II: Express Briefs 53 (2006) 3–7.

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007

[38] X.G. Doukopoulos, G.V. Moustakides, Fast and stable subspace tracking, IEEE Trans. Signal Process. 56 (2008) 1452–1465.

[39] S. Attallah, K. Abed-Meraim, Low-cost adaptive algorithm for noise subspace estimation, Electron. Lett. 38 (2002) 609–611.

[40] S. Bartelmaos, K. Abed-Meraim, Fast minor component extraction using Givens rotations, IEEE Signal Process. Lett. 43 (2007) 1001–1003.

[41] S. Bartelmaos, K. Abed-Meraim, Fast adaptive algorithms for minor component analysis using Householder transformation, Digital Signal Process. 21 (2011) 667–678.

[42] Xiao-Long Zhu, Xian-Da Zhang, Ying Jia, Adaptive RLS Implementation of Non-Negative PCA Algorithm for Blind Source Separation, Lecture Notes in Comput. Sci., vol. 3173, Springer-Verlag, Berlin, Heidelberg, 2004, pp. 761–767.

[43] Y. Bresler, A. Macovski, On the number of signals resolvable by a uniform linear array, IEEE Trans. Acoust. Speech Signal Process. 34 (1986) 1361–1375.

Messaoud Thameri grew up in Msila, Algeria. He received the State Engineering Degree from Ecole Polytechnique (EMP) Algiers, Algeria in 2009. Since 2010, he is working towards his Ph.D. in Signal Processing at Telecom ParisTech, France. His research interests include array and radar signal processing, adaptive filtering and subspace tracking, parameter estimation and asymptotic performance analysis.

Karim Abed-Meraim was born in 1967. He received the State Engineering Degree from Ecole Polytechnique, Paris, France, in 1990, the State Engineering Degree from Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1992, the M.Sc. degree from Paris XI University, Orsay, France, in 1992, and the Ph.D. degree from the Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1995 (in the field of Signal Processing and communications).

From 1995 to 1998, he has been a research staff at the Electrical Engineering Department of the University of Melbourne where he worked on several research project related to “Blind System Identification for Wireless Communications”, “Blind Source Separation”, and “Array Processing for Communications”, respectively.

From 1998 to 2012, he was an Associate Professor at the Signal and Image Processing Department of ENST. Currently, he is a Full Professor at Polytech’Orleans (the university of Orleans), France. His research interests are in signal processing, statistical estimation and communications including system identification, blind source separation, multiuser detection and equalization, detection and estimation, adaptive filtering and tracking, array processing and asymptotic performance analysis. Within his research activity, he has organized and chaired several conferences and workshops and published over 330 scientific papers.

Adel Belouchrani was born in Algiers, Algeria, on May 1967. He received the State Engineering degree in 1991 from Ecole Nationale Polytechnique (ENP), Algiers, Algeria, the M.S. degree in signal processing from the Institut National Polytechnique de Grenoble (INPG), France, in 1992, and the Ph.D. degree in signal and image processing from Télécom Paris (ENST), France, in 1995. He was a Visiting Scholar at the Electrical Engineering and Computer Sciences Department, University of California, Berkeley, from 1995 to 1996. He was with the Department of Electrical and Computer Engineering, Villanova University, Villanova, PA, as a Research Associate from 1996 to 1997. From 1998 to 2005, he has been with the Electrical Engineering Department of ENP as an Associate Professor. Since 2006 he is a Full Professor at ENP. His research interests are in statistical signal processing and (blind) array signal processing with applications in biomedical and communications, time–frequency analysis, time–frequency array signal processing and FPGA implementation of signal processing algorithms. He has published over 160 technical publications including 4 book chapters and 3 patents. His work has been cited over 1300 times (source: ISI web of science).

Please cite this article in press as: M. Thameri et al., Low complexity adaptive algorithms for Principal and Minor Component Analysis, Digital Signal Process. (2012), http://dx.doi.org/10.1016/j.dsp.2012.09.007