

EFFICIENT AND FAST TRACKING ALGORITHM FOR MINOR COMPONENT ANALYSIS

S. Bartelmaos
ENST-Paris, TSI Department
37/39 rue Dareau 75014, Paris
France
bartelma@tsi.enst.fr

K. Abed-Meraim
ENST-Paris, TSI Department
37/39 rue Dareau, 75014 Paris
France
abed@tsi.enst.fr

S. Attallah
NUS university, Dept. of Elec. & Comp. Eng
4 Engineering Drive 3 Singapore 117576
Singapore
attallah@ieee.org

ABSTRACT

In this paper, we propose new adaptive algorithms for the extraction and tracking of the least (minor) eigenvectors of a positive Hermitian covariance matrix. The proposed algorithm is said fast in the sense that its computational cost is of order $O(np)$ flops per iteration where n is the size of the observation vector and $p < n$ is the number of minor eigenvectors we need to estimate.

This algorithm is based on a stochastic gradient technique and a fast orthogonalization procedure that guarantees the algorithm stability and the orthogonality of the weight matrix at each iteration. Despite its low computational cost, the proposed algorithm is quite efficient as shown by simulation experiments and performs better than other existing methods of higher computational complexity.

I. INTRODUCTION

Principal and minor component analysis (PCA and MCA), which are part of the more general principal and minor subspace (PSA and MSA) analysis, are two important problems that are frequently encountered in many information processing fields including communications.

Indeed, Fast estimation and tracking of the principal or minor subspace has been used, for example, in code division multiple access (CDMA) communication where many multiuser detection algorithms are actually subspace-based [1], in the mobile positioning [2], for blind channel equalization [3], etc.

From the computational point of view, we may distinguish between methods requiring $O(n^2p)$, $O(n^2)$, $O(np^2)$, or $O(np)$ operations per update. The wide range of the computational complexity is due to the fact that some algorithms update the complete eigenstructure, with or without the explicit computation of the sample correlation matrix, whereas other ones track only the desired principal or minor subspace [4]. The class of fast subspace tracking methods includes the gradient type algorithms which demand $O(np)$ operations for the gradient-ascent or gradient-descent step and (in the MCA case) additional $O(np^2)$ operations for the orthogonalization of the eigenvector estimates.

In this paper, we propose first an algorithm based on a stochastic gradient technique for tracking minor components. This new algorithm is referred to as MCA-OFRANSH (MCA-Orthogonal Fast Rayleigh's quotient-based Adaptive Noise Subspace using Householder) which applies a fast orthogonalization technique to the (MCA-FRANS) by conserving the

same order $O(np)$ of the computational complexity. This orthogonalization results in a better numerical stability and estimation accuracy. Finally, we propose an appropriate step-size normalization in order to increase the convergence rate. Simulation results are presented to assess the performance of MCA-OFRANSH showing, in particular, that our algorithm reaches the performance of Pastd [5] which requires $O(np^2)$ flops per iteration.

II. ORTHOGONAL MCA-FRANS

Consider the problem of estimating the $p < n$ minor (principal) eigenvectors of the covariance matrix \mathbf{C} . For that, we minimize (maximize) the scalar function [6]

$$J(\mathbf{W}) = \text{tr}(\mathbf{L}\mathbf{W}^H\mathbf{C}\mathbf{W}). \quad (1)$$

where \mathbf{L} is a $p \times p$ positive diagonal matrix containing elements with different weights *i.e.* $\mathbf{L} = \text{diag}(l_1, \dots, l_p)$ with $l_1 > l_2 > \dots > l_p > 0$ and the weight matrix $\mathbf{W} \in \mathbb{C}^{n \times p}$ is an orthogonal matrix of the eigenvectors estimates. This minimization can be achieved by using the gradient-descent technique¹, that is

$$\mathbf{W}(i) = \mathbf{W}(i-1) - \beta \Delta_J(i-1) \quad (2)$$

where β is a positive valued variable step size and the gradient is given by (up to a scalar constant)

$$\Delta_J(i) \propto \mathbf{C}\mathbf{W}\mathbf{L}. \quad (3)$$

Note that, the PCA is treated similarly by changing β into $-\beta$. Now injecting (3) into (2) and replacing \mathbf{C} by its instantaneous estimate at time instant i leads to

$$\mathbf{W}(i) = \mathbf{W}(i-1) - \beta \mathbf{x}(i)\mathbf{z}^H(i) \quad (4)$$

where $\mathbf{z}(i) = \mathbf{L}\mathbf{y}(i)$ and $\mathbf{y}(i) = \mathbf{W}^H(i-1)\mathbf{x}(i)$. The new algorithm is called MCA-FRANS (FRANS is a subspace tracking algorithm based on a similar minimization function [7]). The algorithm is numerically unstable unless orthogonalization of the weight matrix is performed at each step. To this end, we use the following orthogonalization method :

$$\mathbf{W}(i) := \mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{W}(i))^{-1/2}. \quad (5)$$

¹The gradient is calculated as if the entries of \mathbf{W} are free, *i.e.* we relax in a first step the orthogonality constraint on \mathbf{W} .

The fast computation of (5) is obtained thanks to Lemma 1 [8].

Lemma 1: 1- Let \mathbf{R} be a d -rank hermitian matrix spanned by the column vectors $\mathbf{p}_1, \dots, \mathbf{p}_d$, then the eigendecomposition of \mathbf{R} is given by $\mathbf{R} = \mathbf{E}\mathbf{D}\mathbf{E}^H$, where $\mathbf{D} = \text{diag}(\lambda_1 \dots \lambda_d)$ and $\mathbf{E} = [\mathbf{e}_1 \dots \mathbf{e}_d]$ are computed by

$$\begin{aligned} \mathbf{E} &= \mathbf{P}\mathbf{T} \\ \mathbf{P} &\triangleq [\mathbf{p}_1 \dots \mathbf{p}_d] \\ \mathbf{M} &= (\mathbf{P}^H \mathbf{P})^{-1} \mathbf{P}^H \mathbf{R} \mathbf{P} = \mathbf{T} \mathbf{D} \mathbf{T}^{-1} \end{aligned}$$

2- Let $\mathbf{N} = \mathbf{I} + \mathbf{E}\mathbf{D}\mathbf{E}^H$ where \mathbf{E} is orthonormal. Then, an inverse square root of \mathbf{N} is given by

$$\mathbf{N}^{-\frac{1}{2}} = \mathbf{I} + \mathbf{E} \mathbf{D}' \mathbf{E}^H$$

where

$$\mathbf{D}' = \text{diag}\left(\frac{1}{\sqrt{1+\lambda_1}} - 1 \dots \frac{1}{\sqrt{1+\lambda_d}} - 1\right)$$

By applying Lemma 1 to $\mathbf{W}^H(i)\mathbf{W}(i)$ (see appendix for details), we get the Orthogonal version of MCA-FRANS summarized in Table 1. Note that the computational cost of MCA-OFRANS is approximately $4np + O(n)$.

$\mathbf{y}(i)$	$= \mathbf{W}^H(i-1)\mathbf{x}(i)$
$\mathbf{z}(i)$	$= \mathbf{L}\mathbf{y}(i)$
\mathbf{R}	$= -\beta\mathbf{y}(i)\mathbf{z}^H(i) + \mathbf{z}(i)\mathbf{y}^H(i)[- \beta\mathbf{I}_p + \beta^2 \ \mathbf{x}^2(i)\ \mathbf{L}]$
\mathbf{P}	$= [\mathbf{z}(i) \ \mathbf{y}(i)]$
\mathbf{M}	$= (\mathbf{P}^H \mathbf{P})^{-1} \mathbf{P}^H \mathbf{R} \mathbf{P}$
$\text{eig}(\mathbf{M})$	$= \mathbf{T} \text{diag}(\lambda_1, \lambda_2) \mathbf{T}^{-1}$
\mathbf{E}	$= [\mathbf{e}_1 \ \mathbf{e}_2] = \mathbf{P}\mathbf{T}$
Σ	$= \text{diag}\left(\frac{1}{\ \mathbf{e}_1\ }, \frac{1}{\ \mathbf{e}_2\ }\right)$
\mathbf{E}'	$= \mathbf{E}\Sigma \triangleq \begin{bmatrix} \frac{1}{\ \mathbf{e}_1\ } \mathbf{e}_1 \\ \frac{1}{\ \mathbf{e}_2\ } \mathbf{e}_2 \end{bmatrix} \quad \mathbf{T}' = \mathbf{T}\Sigma$
τ_1	$= \frac{1}{\sqrt{1+\lambda_1}} - 1$
τ_2	$= \frac{1}{\sqrt{1+\lambda_2}} - 1$
\mathbf{T}'^{-1}	$= \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$
\mathbf{p}	$= \tau_1 \mathbf{W}(i) \mathbf{e}'_1 - \beta t_{11} (1 + \tau_1) \mathbf{x}(i)$
\mathbf{q}	$= \tau_2 \mathbf{W}(i) \mathbf{e}'_2 - \beta t_{21} (1 + \tau_2) \mathbf{x}(i)$
$\mathbf{W}(i)$	$= \mathbf{W}(i-1) + \mathbf{p}\mathbf{e}'_1{}^H + \mathbf{q}\mathbf{e}'_2{}^H$

Table 1: The MCA-OFRANS.

III. MCA-OFRANS USING HOUSEHOLDER TRANSFORM

In terms of orthogonality errors, the MCA-OFRANS algorithm guarantees the orthogonality of the minor subspace at each iteration. This insures much more stability to the algorithm compared to its original version. However, this improvement doesn't mean absolute stability and the algorithm remains sensitive to numerical rounding errors. This can be shown by theoretical derivation using a similar analysis to that in [9] and is illustrated here by the simulation example of Figure 1. To mitigate the effect of rounding errors, we propose to use the numerically well-behaved Householder orthogonal matrices [10]. To this end, we have proved the following result:

Lemma 2: The updating equation of the weight matrix given by Table 1, can be reformulated as:

$$\mathbf{W}(i) = \mathbf{H}_1(i)\mathbf{H}_2(i)\mathbf{W}(i-1) \quad (6)$$

Where $\mathbf{H}_1(i)$ and $\mathbf{H}_2(i)$ are the Householder transforms given by

$$\begin{aligned} \mathbf{H}_1(i) &\triangleq \mathbf{I} - 2\mathbf{u}(i)\mathbf{u}^H(i). \\ \mathbf{H}_2(i) &\triangleq \mathbf{I} - 2\mathbf{v}(i)\mathbf{v}^H(i). \end{aligned}$$

Where $\mathbf{u}(i)$ (resp. $\mathbf{v}(i)$) is the principal left singular eigenvector of $\mathbf{R} \triangleq \mathbf{W}(i) - \mathbf{W}(i-1)$ (resp. of $\mathbf{Z} = \mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1)$).

The fast computation of $\mathbf{u}(i)$ and $\mathbf{v}(i)$ of order $O(n)$ is given in Table 2 (see appendix for details).

Hence, MCA-OFRANS (H stands for Householder) algorithm can be obtained by adding the equations of Table 2 to the previous table excluding the updating equation of the minor subspace weight matrix. In Table 2, we used the notation $\mathbf{Z}(:, 1)$ to denote the first column vector of \mathbf{Z} , $\mathbf{w}_1(i-1)$ to denote the first column vector of $\mathbf{W}(i-1)$ and \mathbf{e}'_1^* to denote the complex conjugate of \mathbf{e}'_1 . The overall computational cost of MCA-OFRANS is $6np + O(n)$.

\mathbf{Q}	$= [\mathbf{p} \ \mathbf{q}]$
Λ	$= \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} \triangleq \begin{bmatrix} \ \mathbf{e}'_1\ ^2 \ \mathbf{p}\ ^2 + \mathbf{e}'_1{}^H \mathbf{e}'_2 \mathbf{q}^H \mathbf{p} & \mathbf{e}'_2{}^H \mathbf{e}'_1 \ \mathbf{p}\ ^2 + \ \mathbf{e}'_2\ ^2 \mathbf{q}^H \mathbf{p} \\ \ \mathbf{e}'_1\ ^2 \mathbf{p}^H \mathbf{q} + \mathbf{e}'_1{}^H \mathbf{e}_2 \ \mathbf{q}\ ^2 & \mathbf{e}_2{}^H \mathbf{e}_1 \mathbf{p}^H \mathbf{q} + \ \mathbf{e}_2\ ^2 \ \mathbf{q}\ ^2 \end{bmatrix}$
\mathbf{Q}_2	$\triangleq [\lambda_{11} \mathbf{p} + \lambda_{12} \mathbf{q} \ \lambda_{21} \mathbf{p} + \lambda_{22} \mathbf{q}]$
\mathbf{R}_q	$= (\mathbf{Q}^H \mathbf{Q})^{-1} \mathbf{Q}^H \mathbf{Q}_2$
$\text{eig}(\mathbf{R}_q)$	$= \mathbf{T}_1 \text{diag}(\lambda_{q1}, \lambda_{q2}) \mathbf{T}_1^{-1}$
\mathbf{H}	$= [\mathbf{h}_1 \ \mathbf{h}_2] = \mathbf{Q} \mathbf{T}_1$
\mathbf{u}	$= \frac{\mathbf{h}_1}{\ \mathbf{h}_1\ }$
$\mathbf{Z}(:, 1)$	$= 2\mathbf{u}(\mathbf{u}^H \mathbf{p} \mathbf{e}'_1{}^*(1) + \mathbf{u}^H \mathbf{p} \mathbf{e}'_2{}^*(1) + \mathbf{u}^H \mathbf{w}_1(i-1)) - \mathbf{p} \mathbf{e}'_1{}^*(1) - \mathbf{q} \mathbf{e}'_2{}^*(1)$
\mathbf{v}	$= \frac{\mathbf{Z}(:, 1)}{\ \mathbf{Z}(:, 1)\ }$
$\mathbf{W}(i)$	$= (\mathbf{I} - 2\mathbf{u}\mathbf{u}^H)(\mathbf{I} - 2\mathbf{v}\mathbf{v}^H)\mathbf{W}(i-1)$

Table 2: The MCA-OFRANS.

The MCA-OFRANS becomes numerically very stable, as illustrated by our simulation results.

IV. NORMALIZATION OF THE STEP SIZE

We present here a normalized version of the new gradient-descent based algorithm in order to have a faster convergence. To this end, we inject (2) into (1) and using a variable step-size $\beta(i)$, we can write

$$\begin{aligned} \mathbf{J}(i) &= \text{tr}(\mathbf{L}(\mathbf{W}^H(i-1) - \beta(i-1)\Delta_J^H(i-1))\mathbf{C} \\ &\quad (\mathbf{W}(i-1) - \beta(i-1)\Delta_J(i-1))) \\ &= \mathbf{J}(i-1) - 2\beta(i-1)\text{tr}(\mathbf{W}^H(i-1)\mathbf{C}\Delta_J(i-1)\mathbf{L}) \\ &\quad + \beta^2(i-1)\text{tr}(\mathbf{L}\Delta_J^H(i-1)\mathbf{C}\Delta_J(i-1)). \end{aligned} \quad (7)$$

This means that $\mathbf{J}(i)$ is a quadratic function of $\beta(i-1)$ which has a global minimum. Hence, the optimal step-size can be

found by setting $\frac{\partial \mathbf{J}(i)}{\partial \beta(i-1)}$ to zero

$$\beta_{opt}(i-1) = \frac{\text{tr}(\mathbf{W}^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1)\mathbf{L})}{\text{tr}(\mathbf{L}\mathbf{\Delta}_J^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))}. \quad (8)$$

By replacing $\mathbf{\Delta}_J(i-1)$ and \mathbf{C} by their instantaneous estimates, we obtain finally

$$\begin{aligned} \beta_{opt}(i) &= \frac{\text{tr}(\mathbf{L}^2 \mathbf{y}(i) \mathbf{y}^H(i))}{\|\mathbf{x}(i)\|^2 \text{tr}(\mathbf{L}^3 \mathbf{y}(i) \mathbf{y}^H(i))} \\ &= \frac{\|\mathbf{z}(i)\|_{\mathbf{I}}^2}{\|\mathbf{x}(i)\|^2 \|\mathbf{z}(i)\|_{\mathbf{L}}^2} \end{aligned} \quad (9)$$

where $\|\mathbf{z}\|_{\mathbf{L}}^2 \triangleq \mathbf{z}^H \mathbf{L} \mathbf{z}$.

A. Convergence analysis

Using equations (7) and (8) we get,

$$\begin{aligned} \mathbf{J}(i) &= \mathbf{J}(i-1) - 2 \frac{\text{tr}^2(\mathbf{L}\mathbf{W}^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))}{\text{tr}(\mathbf{L}\mathbf{\Delta}_J^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))} + \\ &\quad \frac{\text{tr}^2(\mathbf{L}\mathbf{W}^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))}{\text{tr}(\mathbf{L}\mathbf{\Delta}_J^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))} \\ \Rightarrow \mathbf{J}(i) - \mathbf{J}(i-1) &= - \frac{\text{tr}^2(\mathbf{L}\mathbf{W}^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))}{\text{tr}(\mathbf{L}\mathbf{\Delta}_J^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1))}. \end{aligned}$$

We can easily notice that

$$\text{tr}^2(\mathbf{L}\mathbf{W}^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1)) \geq 0, \forall \mathbf{W}, \mathbf{C}. \quad (10)$$

and, assuming \mathbf{C} to be strictly positive definite, we have

$$\text{tr}(\mathbf{L}\mathbf{\Delta}_J^H(i-1)\mathbf{C}\mathbf{\Delta}_J(i-1)) > 0, \forall \mathbf{\Delta}_J(i-1) \neq 0. \quad (11)$$

Therefore

$$\mathbf{J}(i) - \mathbf{J}(i-1) = -\epsilon(i) \text{ where } \epsilon \geq 0. \quad (12)$$

The mean square error (MSE) will decrease at each iteration (minimization) by a decrement of $\epsilon(i-1)$. Therefore, the larger the decrement, the faster the convergence is. Since $\mathbf{J}(\beta)$ has a global minimum, the step-size $\beta_{opt}(i)$ will lead to the maximum decrement $\epsilon(i)$ at each iteration. Note that the MSE will decrease even if we replace \mathbf{C} and the gradient by their corresponding estimates. Finally, we can write

$$\hat{\beta}_{opt} = \beta \frac{\|\mathbf{z}(i)\|_{\mathbf{I}}^2}{\|\mathbf{x}(i)\|^2 \|\mathbf{z}(i)\|_{\mathbf{L}}^2 + \gamma} \quad (13)$$

where β and γ are two positive constants ($0 < \beta$ and $\gamma < 1$) which help improve the numerical stability of the algorithm [11].

V. SIMULATIONS

To assess the performance of our algorithm, we calculate the ensemble average of the performance factors $\rho(i) = \frac{1}{p \cdot r_0} \sum_{r=1}^{r_0} \|\mathbf{W}_r(i) - \mathbf{E}_2\|^2$ and $\eta(i) = \frac{1}{r_0} \sum_{r=1}^{r_0} \|\mathbf{W}_r^H(i) \mathbf{W}_r(i) - \mathbf{I}\|^2$, where the number of algorithm runs is set to $r_0 = 50$, r indicates that the associated variable depends on the particular

run. \mathbf{E}_2 is the $n \times p$ matrix of the p minor eigenvectors and $\mathbf{W}(i)$ represents the $n \times p$ matrix of the p estimated principal eigenvectors.

The first performance index ρ measures the averaged estimation accuracy of the eigenvectors while the second performance index η measures the orthogonality of the weight matrix. Note that, as the eigenvectors are estimated up to a phase indeterminacy, we remove this ambiguity (by forcing the first entry of each eigenvector to be positive) before the comparison in the performance factor η .

In the simulation experiment, we have considered for (Fig.1, Fig.2 and Fig.3), an *iid* sequence of n -dimensional random vectors $\mathbf{x}(i)$ with $n = 10$. The random sequence is generated using a zero mean Gaussian-distribution with positive definite covariance matrix \mathbf{C} that is generated randomly at each run.

In Fig.1, we extract the $p = 3$ minor eigenvectors of \mathbf{C} using the OFRANS and the OFRANSH methods with a step size $\beta = 0.01$. As we can see, OFRANSH is numerically stable.

In Fig.2, we compare the normalized version of MCA-OFRANSH (variable step-size (13) with: $\beta = 0.5$ and $\gamma = 0.4$) with its initial version with constant step-size $\beta = 0.01$. Clearly the step-size normalization improves the convergence rate of the algorithm, but slightly increases the steady state error.

For Fig.3, we compare the performance of OFRANSH ($\beta = 0.5$, $\gamma = 0.4$) with the Pastd method with a forgetting factor $\alpha = 0.99$. A similar comparison is given in Fig.4 but using the simulation set-up proposed in [12]. The latter corresponds to choosing ($n = 4$, $p = 2$, $\beta = 0.4$, $\gamma = 0.4$) and $\mathbf{x}(i)$ a sequence of independent jointly-Gaussian random vectors with covariance matrix

$$\mathbf{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix} \quad (14)$$

From the results of Fig.3 and Fig.4, we can see that our algorithm reaches the performance of Pastd algorithm that requires $O(np^2)$ flops per iteration instead of $O(np)$ for our algorithm.

VI. CONCLUSION

In this paper, we present a new algorithm with low computational complexity of order $O(np)$ for minor component analysis. The proposed algorithm is based on an existing method (FRANS in [7]) to which we have introduced a fast orthogonalization of the weight matrix plus an optimal normalization of the step-size.

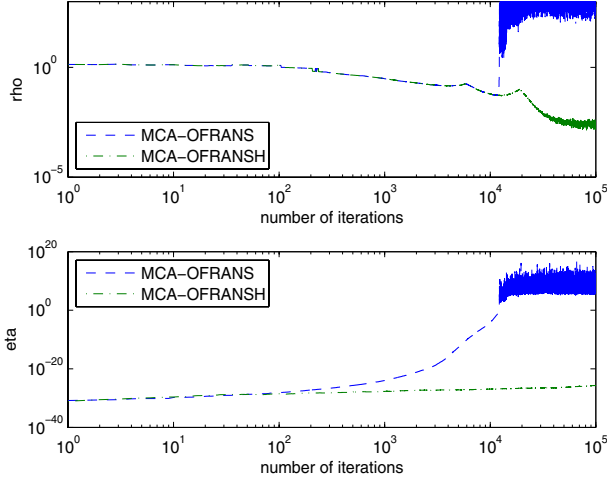
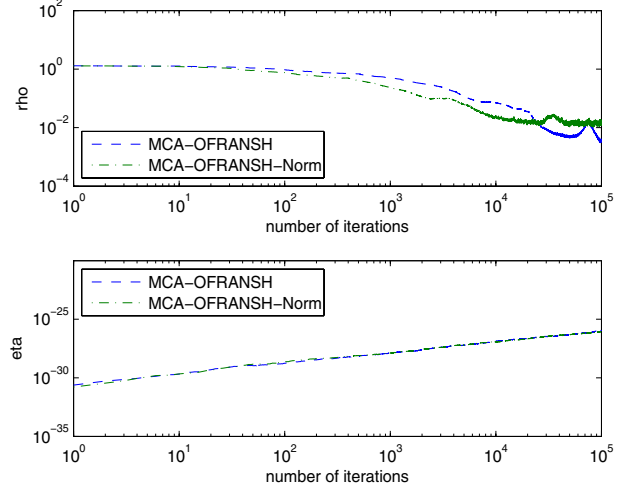
The resulting algorithm OFRANSH achieves the MCA with a performance comparable to that of Pastd, a more expensive algorithm of complexity $O(np^2)$.

We should note that $\eta(i)$ is slowly increasing as shown in simulation results, it means that the MCA-OFRANSH algorithm is not absolutely stable, one can guarantee the orthogonality by applying periodically the Gram-shmidt orthogonalization.

VII. APPENDIX

A. Orthogonalization of FRANS algorithm

To compute (5), we use the updating equation of $\mathbf{W}(i)$. Keeping in mind that $\mathbf{W}(i-1)$ is now an orthogonal matrix, we


 Figure 1: Performance of MCA-OFRANS and MCA-OFRANSH ($p = 3, n = 10$).

 Figure 2: Performance of MCA-OFRANSH-Norm and MCA-OFRANSH ($p = 3, n = 10$).

have (we omit the index i for simplicity)

$$\begin{aligned} \mathbf{N} &\triangleq \mathbf{W}^H(i)\mathbf{W}(i) \\ &= \mathbf{I} - \beta\mathbf{y}\mathbf{z}^H - \beta\mathbf{z}\mathbf{y}^H + \beta^2 \|\mathbf{x}\|^2 \mathbf{z}\mathbf{z}^H \\ &= \mathbf{I} + \mathbf{R} \end{aligned}$$

\mathbf{R} is a rank-2 Hermitian matrix and hence one can apply *Lemma 1*'s result to obtain its fast eigendecomposition and to compute the inverse square root of \mathbf{N} according to the following steps:

$$\begin{aligned} \mathbf{P} &= [\mathbf{z} \ \mathbf{y}] \\ \mathbf{M} &= (\mathbf{P}^H\mathbf{P})^{-1}\mathbf{P}^H\mathbf{R}\mathbf{P} \\ \mathbf{M} &= \mathbf{T}\text{diag}(\lambda_1, \lambda_2)\mathbf{T}^{-1} \\ \mathbf{E} &= \mathbf{P}\mathbf{T} = [\mathbf{e}_1 \ \mathbf{e}_2]. \end{aligned}$$

As the eigenvectors \mathbf{T} are computed up to scalar factors, one needs to normalize the columns of \mathbf{E} to force it to be unitary, *i.e.*

$$\begin{aligned} \Sigma &= \text{diag}\left(\frac{1}{\|\mathbf{e}_1\|}, \frac{1}{\|\mathbf{e}_2\|}\right) \\ \mathbf{E}' &= \mathbf{E}\Sigma \text{ and } \mathbf{T}' = \mathbf{T}\Sigma. \end{aligned}$$

Now, according to *Lemma 1*, we have

$$\mathbf{N}^{-\frac{1}{2}} = \mathbf{I} + \mathbf{E}'\mathbf{D}'\mathbf{E}'^H \quad (15)$$

where

$$\begin{aligned} \mathbf{D}' &= \text{diag}\left(\frac{1}{\sqrt{1+\lambda_1}} - 1, \frac{1}{\sqrt{1+\lambda_2}} - 1\right) \\ &= \text{diag}(\tau_1, \tau_2) \end{aligned} \quad (16)$$

By substituting (15) into (5) we obtain

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{W}(i-1) + \tau_1 \mathbf{W}(i-1)\mathbf{e}_1'\mathbf{e}_1'^H + \\ &\quad \tau_2 \mathbf{W}(i-1)\mathbf{e}_2'\mathbf{e}_2'^H - \beta\mathbf{x}\mathbf{z}^H - \beta\tau_1\mathbf{x}\mathbf{z}^H\mathbf{e}_1'\mathbf{e}_1'^H \\ &\quad - \beta\tau_2\mathbf{x}\mathbf{z}^H\mathbf{e}_2'\mathbf{e}_2'^H \end{aligned} \quad (17)$$

Using the orthonormality of \mathbf{E}' (*i.e.*, $\mathbf{E}'^H\mathbf{E}' = \mathbf{I}$) we obtain $(\mathbf{T}'^{-1} = \mathbf{E}'^H\mathbf{P})$. Hence, we can write

$$\begin{aligned} \mathbf{T}'^{-H} &= \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}^H \\ &= \begin{bmatrix} \mathbf{z}^H\mathbf{e}_1' & \mathbf{z}^H\mathbf{e}_2' \\ \mathbf{y}^H\mathbf{e}_1' & \mathbf{y}^H\mathbf{e}_2' \end{bmatrix} \end{aligned} \quad (18)$$

Also, by developing $\mathbf{P} = \mathbf{E}'\mathbf{T}'^{-1}$ we get

$$\mathbf{z} = t_{11}\mathbf{e}_1' + t_{21}\mathbf{e}_2'. \quad (19)$$

Finally, if we replace the results obtained in (18) and (19) into (17) we obtain

$$\mathbf{W}(i) = \mathbf{W}(i-1) + \mathbf{p}\mathbf{e}_1'^H + \mathbf{q}\mathbf{e}_2'^H \quad (20)$$

Where

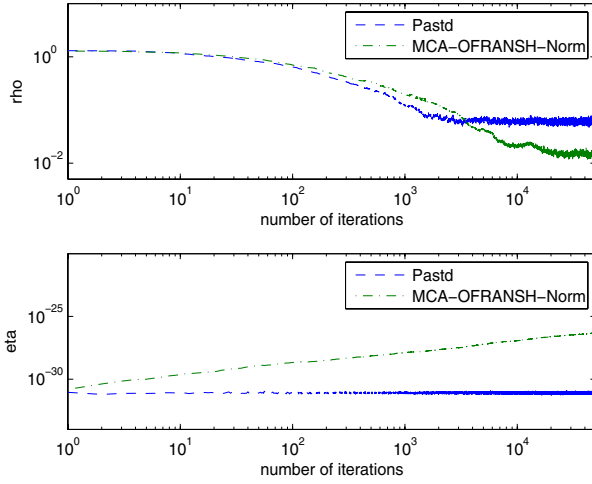
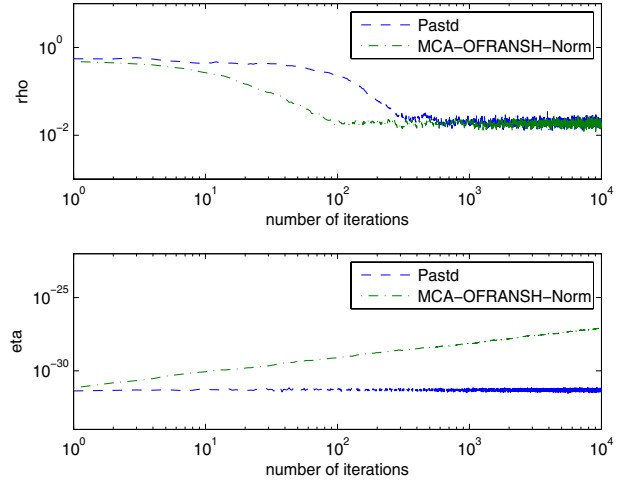
$$\mathbf{p} = \tau_1 \mathbf{W}(i-1)\mathbf{e}_1' - \beta t_{11}(1 + \tau_1)\mathbf{x}(i) \quad (21)$$

$$\mathbf{q} = \tau_2 \mathbf{W}(i-1)\mathbf{e}_2' - \beta t_{21}(1 + \tau_2)\mathbf{x}(i) \quad (22)$$

B. Householder

As stated by *Lemma 2*, \mathbf{u} is calculated as the principal left singular eigenvector of

$$\mathbf{Q}' = \mathbf{W}(i) - \mathbf{W}(i-1) \quad (23)$$


 Figure 3: Performance of MCA-OFRANSH and Pastd ($p = 3, n = 10$).

 Figure 4: Performance of MCA-OFRANSH and Pastd ($p = 2, n = 4$).

where $\mathbf{Q}' = \mathbf{Q}\mathbf{E}^H = \mathbf{p}\mathbf{e}_1'^H + \mathbf{q}\mathbf{e}_2'^H$.

Equivalently, \mathbf{u} can be seen as the principal eigenvector of the rank 2 Hermitian matrix

$$\mathbf{R} = \mathbf{Q}'\mathbf{Q}'^H$$

This can be done by using *Lemma 1* results as follows:

$$\begin{aligned} \mathbf{Q} &= [\mathbf{p} \ \mathbf{q}] \\ \mathbf{R}_2 &= (\mathbf{Q}^H\mathbf{Q})^{-1}\mathbf{Q}^H\mathbf{R}\mathbf{Q} = (\mathbf{Q}^H\mathbf{Q})^{-1}\mathbf{Q}^H\mathbf{Q}_2 \end{aligned}$$

where

$$\begin{aligned} \mathbf{Q}_2 &= [\lambda_{11}\mathbf{p} + \lambda_{12}\mathbf{q} \quad \lambda_{21}\mathbf{p} + \lambda_{22}\mathbf{q}] \\ \Lambda &= \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \\ &= \begin{bmatrix} \|\mathbf{e}_1'\|^2\|\mathbf{p}\|^2 + \mathbf{e}_1'^H\mathbf{e}_2'\mathbf{q}^H\mathbf{p} & \mathbf{e}_2'^H\mathbf{e}_1'\|\mathbf{p}\|^2 + \|\mathbf{e}_2'\|^2\mathbf{q}^H\mathbf{p} \\ \|\mathbf{e}_1'\|^2\mathbf{p}^H\mathbf{q} + \mathbf{e}_1'^H\mathbf{e}_2'\|\mathbf{q}\|^2 & \mathbf{e}_2'^H\mathbf{e}_1'\mathbf{p}^H\mathbf{q} + \|\mathbf{e}_2'\|^2\|\mathbf{q}\|^2 \end{bmatrix}. \end{aligned}$$

The principal eigenvectors of \mathbf{R} are given by $\mathbf{H} = \mathbf{Q}\mathbf{T}_1 = [\mathbf{h}_1 \ \mathbf{h}_2]$, \mathbf{T}_1 being the eigenvector matrix of \mathbf{R}_2 , i.e.

$$\text{eig}(\mathbf{R}_2) = \mathbf{T}_1 \text{diag}(\lambda_{q_1}, \lambda_{q_2}) \mathbf{T}_1^{-1}.$$

Note that both principal eigenvectors of \mathbf{R} can do the job, i.e. one can choose either $\mathbf{u} = \frac{\mathbf{h}_1}{\|\mathbf{h}_1\|}$ or $\mathbf{u} = \frac{\mathbf{h}_2}{\|\mathbf{h}_2\|}$. Now, once \mathbf{u} is computed, we can observe that

$$\mathbf{Z} = \mathbf{H}_1\mathbf{W}(i) - \mathbf{W}(i-1) = -2\mathbf{v}\mathbf{v}^H\mathbf{W}(i-1)$$

is a rank-1 matrix. All column vectors of the previous matrix are equal to \mathbf{v} (up to scalar constant). Hence, it is sufficient to compute only its first column vector $\mathbf{Z}(:,1)$ and take \mathbf{v} as its normalized version, i.e. $\mathbf{v} = \frac{\mathbf{Z}(:,1)}{\|\mathbf{Z}(:,1)\|}$.

This leads to the updating equation in Table 2.

REFERENCES

- [1] X. Wang, H. V. Poor, P. V. 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, "A new blind adaptive mmse equalizer for mimo systems," *PIMRC'05*, Septembre 2005.
- [4] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc of the IEEE*, vol. 78, no. 8, pp. 1327–1343, Aug 1990.
- [5] H. Sakai and K. Shmizu, "A new adaptative algorithm for minor component analysis," *Signal Processing* 71, pp. 301–308, 1998.
- [6] L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, vol. 6, pp. 627–648, 1993.
- [7] S. Attallah and K. Abed-Meraim, "Low-cost adaptive algorithm for noise subspace estimation," *Electronics Letters*, vol. 38 no. 12, pp. 609–611, June 2002.
- [8] K. Abed-Meraim, A. Chkeif, Y. Hua, and S. Attallah, "On a class of orthonormal algorithms for principal and minor subspace tracking," *Journal of VLSI Signal Processing Systems (invited paper)*, 2001.
- [9] X. Doukopoulos, "Power techniques for blind channel estimation in wireless communication systems," *Phd thesis*, pp. 125–130, October 2004.
- [10] G. Golub and C. V. Loan, "Matrix computations," *Baltimore, MD: Johns Hopkins Univ. Press*, pp. 205–213, 1996.
- [11] S. Haykin, "Adaptive filters," *Englewood Cliffs, Prentice-Hall, NJ*, March 1995.
- [12] S. C. Douglas, S.-Y. Kung, and S.-I. Amari, "A self-stabilized minor subspace rule," *Signal processing Lett.*, vol. 5, pp. 328–330, Dec. 1998.