PRINCIPAL AND MINOR SUBSPACE TRACKING: ALGORITHMS & STABILITY ANALYSIS

S. Bartelmaos & K. Abed-Meraim

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

ABSTRACT

We consider the problem of tracking the minor or principal subspace of a positive Hermitian covariance matrix. We first propose a fast and numerically robust implementation of Oja algorithm (FOOja: Fast Orthogonal Oja). The latter 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 or principal eigenvectors we need to estimate. FOOja guarantees the orthogonality of the weight matrix at each iteration. Moreover, this algorithm is analyzed and compared with two other fast algorithms (OOjaH and FDPM) with respect to their numerical stability. Simulation results highlights the relatively good stability behavior of FOOja.

1. INTRODUCTION

Principal and Minor subspace (PSA and MSA) analysis, are two important problems that are frequently encountered in many information processing fields such as: telecommunication, adaptive filtering, direction of arrivals estimation, antenna array processing, etc. Many subspace tracking algorithms exist in the literature that can be classified according to their computational complexity. Since usually $p << n$, schemes requiring $O(n^2)$ or $O(n^2p)$ operations will be classified as high complexity; algorithms with complexity $O(np^2)$ as medium complexity, finally algorithmic schemes requiring only $O(np)$ operations are said of low complexity. 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. For example, the parallel method proposed by Moonen et al [1], which updates the SVD by interlaced QR triangularization and Jacobi rotations, requires $O(n^2)$ operations. The gradient (Oja) type algorithms track either the principal or the minor subspace. They demand $O(np)$ operations for the gradient-ascent or gradient-descent step and additional $O(np^2)$ operations for the orthogonalization of the eigenvector estimates.

The great majority of articles addressing the problem of subspace tracking focuses on the signal subspace. The literature intended for the noise (minor) subspace is unfortunately very limited. In our case, we focus mainly on the minor subspace analysis.

Besides the computational cost, the algorithm numerical stability is of crucial importance. In [2], the authors claimed that their FDPM algorithm is the only minor subspace tracking algorithm of complexity $O(np)$ that is absolutely stable. Hence, we propose a new algorithm referred to as FOOja for (Fast Orthogonal Oja) that behaves better than FDPM and the Householder Oja (OOjaH) in terms of tracking performance & stability.

2. MINOR SUBSPACE EXTRACTION

Let $\mathbf{r}(k)$ be a sequence of $n \times 1$ random vectors with covariance matrix $\mathbf{C} = E[\mathbf{r}(k)\mathbf{r}^H(k)]$. Consider the problem of extracting the minor subspace spanned by the sequence, of dimension $p < n$, assumed to be the span of the p minor eigenvectors of the covariance matrix. To tackle this problem, several subspace extraction algorithms have been proposed in the literature [3]-[4]. Here we recall briefly two of the most efficient MSA algorithms of order $O(np)$, namely the OOjaH and the FDPM, that are used later for performance comparison.

2.1. Orthogonal Oja

The minor subspace extraction algorithm by Oja et al. [5] can be expressed as

$$\begin{aligned} \mathbf{W}(i) &= \mathbf{W}(i-1) - \beta(\mathbf{r}(i)\mathbf{y}^H(i) - \mathbf{z}(i)\mathbf{y}^H(i)) \\ &= \mathbf{W}(i-1) - \beta\mathbf{p}(i)\mathbf{y}^H(i) \end{aligned} \tag{1}$$

Where $\mathbf{W}(i) \in \mathbf{C}^{n\times p}$ is the minor subspace estimate, $\mathbf{y}(i) \triangleq \mathbf{W}^H(i-1)\mathbf{r}(i)$, $\mathbf{z}(i) \triangleq \mathbf{W}(i-1)\mathbf{y}(i)$, $\mathbf{p}(i) \triangleq \mathbf{r}(i) - \mathbf{z}(i)$, and $\beta > 0$ is a learning parameter. Reversing the sign of the adaptive gain, that is replacing $-\beta$ in (1) by $+\beta$, yields a principal subspace extraction algorithm. Equation (1) represents the updating of the weight matrix $\mathbf{W}(i)$ at the i-th iteration. For MSA, Oja is known to diverge. Recently, an orthogonalization step of the weight matrix has been introduced to the Oja algorithm in order to have more stability. The new algorithm is called OOjaH [6] (H stands for Householder) and is summarized in Table 1.

$$\begin{array}{|rcl|}\hline \mathbf{y}(i) &=& \mathbf{W}^H(i-1)\mathbf{r}(i) \\ \mathbf{z}(i) &=& \mathbf{W}(i-1)\mathbf{y}(i) \\ \mathbf{p}(i) &=& \mathbf{r}(i)-\mathbf{z}(i) \\ \phi(i) &=& \frac{1}{\sqrt{1+\beta^2\|\mathbf{p}(i)\|^2\|\mathbf{y}(i)\|^2}} \\ \tau(i) &=& \frac{\phi(i)-1}{\|\mathbf{y}(i)\|^2} \\ \bar{\mathbf{p}}(i) &=& -\tau(i)\mathbf{z}(i)/\beta+\phi(i)\mathbf{p}(i) \\ \mathbf{u}(i) &=& \bar{\mathbf{p}}(i)/\|\bar{\mathbf{p}}(i)\| \\ \mathbf{v}(i) &=& \mathbf{W}^H(i-1)\mathbf{u}(i) \\ \mathbf{W}(i) &=& \mathbf{W}(i-1)-2\mathbf{u}(i)\mathbf{v}^H(i). \\ \hline \end{array}$$

Table 1. The OOjaH.

2.2. FDPM algorithm

the Fast Data Projection Method (FDPM) is a recently proposed fast algorithm supposed to be the most efficient minor subspace tracking algorithm of complexity $O(np)$ [2]. It is summarized in Table 2 ($\mathbf{e}_1 = [1\ 0\ldots 0]^H$).

$$\begin{array}{|rcl|}\hline \mathbf{y}(i) &=& \mathbf{W}^H(i-1)\mathbf{r}(i) \\ \mathbf{T}(i) &=& \mathbf{W}(i-1)-\frac{\beta}{\|\mathbf{r}(i)\|^2}\mathbf{r}(i)\mathbf{y}^H(i) \\ \mathbf{a}(i) &=& \mathbf{y}(i)-\|\mathbf{y}(i)\|\,\mathbf{e}_1 \\ \mathbf{Z}(i) &=& \mathbf{T}(i)-\frac{2}{\|\mathbf{a}\|^2}\left[\mathbf{T}(i)\mathbf{a}(i)\right]\mathbf{a}^H(i) \\ \mathbf{D} &=& \left(diag\left(\mathbf{Z}^H(i)\mathbf{Z}(i)\right)\right)^{\frac{-1}{2}} \\ \mathbf{W}(i) &=& \mathbf{Z}(i)\mathbf{D}(i) \\ \hline \end{array}$$

Table 2. The FDPM.

3. FAST ORTHOGONAL OJA

In this paper, we propose an alternative approximate orthonormalization procedure for Oja. The resulting algorithm is numerically 'more stable' as will be shown by simulations. Our algorithm consists of (1) plus a fast orthonormalization procedure of the weight matrix at each iteration:

$$\begin{aligned} \mathbf{T}(i) &= \mathbf{W}(i-1)\pm\beta\mathbf{p}(i)\mathbf{y}^H(i) \\ \mathbf{W}(i) &= orthonormal\{\mathbf{T}(i)\} \end{aligned} \tag{2}$$

Let us rewrite (2) in the following form:

$$\mathbf{W}(i)=\mathbf{T}(i)\mathbf{H}(i). \tag{3}$$

Where the matrix $\mathbf{H}(i)$ performs an orthogonalization procedure. One way to find possible structures of $\mathbf{H}(i)$ is to consider the product $\mathbf{W}^H(i)\mathbf{W}(i)$. Therefore, forming $\mathbf{W}^H(i)\mathbf{W}(i)$, assuming that $\mathbf{W}^H(i-1)\mathbf{W}(i-1)$ is orthonormal i.e. ($\mathbf{W}^H(i-1)\mathbf{W}(i-1)=\mathbf{I}$), we get:

$$\mathbf{W}^H(i)\mathbf{W}(i)=\mathbf{H}^H(i)\left(\mathbf{I}+\beta^2\left\|\mathbf{p}^2\right\|\mathbf{y}(i)\mathbf{y}^H(i)\right)\mathbf{H}(i) \tag{4}$$

The OOjaH algorithm proposes to use as $\mathbf{H}(i)$ the inverse square root of the matrix $\mathbf{I}+\beta^2\left\|\mathbf{p}^2\right\|\mathbf{y}(i)\mathbf{y}^H(i)$ [6]. However, such an approach diverges slowly from orthonormality in the case of noise subspace tracking. In our algorithm, we are going to proceed differently. Following the same spirit as in [2], we seek a matrix $\mathbf{H}(i)$ in order to make the product in (4) a diagonal matrix and not exactly equal to the identity. In other words we look for a matrix $\mathbf{H}(i)$ that orthogonalizes $\mathbf{T}(i)$ and not directly orthonormalizes it.
To find $\mathbf{H}(i)$ we use Householder reflection properties [7] that can be resumed in the following lemma
Lemma *: Let $\mathbf{v}\in R^n$ be a nonzero vector. An $n\times n$ matrix $\mathbf{H}$ of the form*

$$\mathbf{H}=\mathbf{I}-\frac{2}{\mathbf{v}^H\mathbf{v}}\mathbf{v}\mathbf{v}^H$$

is called Householder reflection. If a vector $\mathbf{x}$ is multiplied by $\mathbf{H}$, then it is reflected in the hyperplan span $\{\mathbf{v}\}^\perp$.

Applying this lemma in order to have $\mathbf{H}(i)\mathbf{y}(i)$ multiple of $\mathbf{e}_1$, we obtain:

$$\mathbf{H}(i)=\mathbf{I}-\frac{2}{\mathbf{a}^H(i)\mathbf{a}(i)}\mathbf{a}(i)\mathbf{a}^H(i) \tag{5}$$

with $\mathbf{a}(i)=\mathbf{y}(i)-\|\mathbf{y}(i)\|\,\mathbf{e}_1$. It is easy to verify that Householder matrices are hermitian and orthogonal. The product in (4) becomes the following diagonal matrix:

$$\mathbf{W}^H(i)\mathbf{W}(i)=\mathbf{I}+\beta^2\left\|\mathbf{p}^2\right\|\|\mathbf{y}(i)\|^2\,\mathbf{e}_1\mathbf{e}_1^H. \tag{6}$$

Finally we normalize (3), in order to have an orthonormal matrix. The FOOja is summarized in Table 3. The FOOja is numerically more stable than OOjaH and FDPM and has the best performance among minor subspace tracking algorithms having a computational cost of order $O(np)$ as illustrated by the simulation results.

$$\begin{array}{|rcl|}\hline \mathbf{y}(i) &=& \mathbf{W}^H(i-1)\mathbf{r}(i) \\ \mathbf{z}(i) &=& \mathbf{W}(i-1)\mathbf{y}(i) \\ \mathbf{p}(i) &=& \mathbf{r}(i)-\mathbf{z}(i) \\ \mathbf{T}(i) &=& \mathbf{W}(i-1)-\beta\mathbf{p}(i)\mathbf{y}^H(i) \\ \mathbf{a}(i) &=& \mathbf{y}(i)-\|\mathbf{y}(i)\|\,\mathbf{e}_1 \\ \mathbf{Z}(i) &=& \mathbf{T}(i)-\frac{2}{\|\mathbf{a}\|^2}\left[\mathbf{T}(i)\mathbf{a}(i)\right]\mathbf{a}^H(i) \\ \mathbf{D} &=& \left(diag\left(\mathbf{Z}^H(i)\mathbf{Z}(i)\right)\right)^{\frac{-1}{2}} \\ \mathbf{W}(i) &=& \mathbf{Z}(i)\mathbf{D}(i) \\ \hline \end{array}$$

Table 3. The FOOja.

Remarks:

- As we can observe in (6), only the first column vector of $\mathbf{W}(i)$ needs to be normalized. However a theoretical stability analysis for FOOja and FDPM algorithms proves that such approach makes the algorithm numerically instable as shown in Fig. 3. We observed in our experiments that normalizing all the column vectors of $\mathbf{W}(i)$ leads to better stability behavior (see Figures 1 and 2). For principal subspace analysis it is not necessary to normalize all the column vector of $\mathbf{W}(i)$ which reduces slightly the computational cost.
- The updating equations of FDPM and FOOja in Tables 2 and 3 are for the real case. In the complex case, we need to replace $\mathbf{a}(i)$ in Tables 2 and 3 by

$$\mathbf{a}(i)=\ \mathbf{y}(i)-\|\mathbf{y}(i)\|\,e^{j\left(angle\left(\mathbf{e}_1^H\mathbf{y}(i)\right)\right)}\mathbf{e}_1$$

where $angle(x)$ represents the phase argument of a complex x.

- We should note that FDPM and FOOja have a main advantage compared to other MSA algorithms, in the sense that the weight matrix retrieves its orthogonality when lost thanks to the particular orthogonalization scheme that is used in these two algorithms (see Fig. 2).

4. STABILITY ANALYSIS

4.1. Numerical stability of FOOja

Here we analyze the numerical stability of FOOja algorithm when the first column vector of $\mathbf{W}$(i) is normalized. To examine the numerical stability we focus on the deviation of the algorithm from orthonormality. Let us first consider the matrix $\mathbf{W}^H(i)\mathbf{W}(i)$ which, in the ideal case should be equal to the identity. Using the equations of Table 2, and after some straightforward manipulations we get:

$$\begin{aligned}\mathbf{W}^H(i)\mathbf{W}(i) = \ & \mathbf{D}(i)[\mathbf{H}(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\mathbf{H}(i) \\ & +\rho\left\|\mathbf{y}\right\|^2\mathbf{e}_1\mathbf{e}_1^H + \beta(\mathbf{H}(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\mathbf{y}(i)\mathbf{y}^H(i)\mathbf{H}(i) \\ & +\mathbf{H}(i)\mathbf{y}(i)\mathbf{y}^H(i)\mathbf{W}^H(i-1)\mathbf{W}(i-1)\mathbf{H}(i))]\mathbf{D}(i)\end{aligned}$$

where $\rho = -2\beta + \beta^2\left\|\mathbf{p}(i)\right\|^2$.
Let ε(i) be the deviation from orthonormality due to numerical rounding errors i.e. $\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{I} + \varepsilon(i)$, we obtain

$$\begin{aligned}\varepsilon(i) = \ & \mathbf{D}(i)[\mathbf{H}(i)\varepsilon(i-1)\mathbf{H}(i) + \mathbf{I} + \beta^2\left\|\mathbf{p}(i)\right\|^2\left\|\mathbf{y}\right\|^2\mathbf{e}_1\mathbf{e}_1^H \\ & +\beta(\mathbf{H}(i)\varepsilon(i-1)\mathbf{\Omega}(i)\mathbf{H}(i) \\ & +\mathbf{H}(i)\mathbf{\Omega}(i)\varepsilon(i-1)\mathbf{H}(i))]\mathbf{D}(i) - \mathbf{I}.\end{aligned} \quad (7)$$

Where $\mathbf{\Omega}(i) \triangleq \mathbf{y}(i)\mathbf{y}^H(i)$. Since $\mathbf{D}\left(\mathbf{I} + \beta^2\left\|\mathbf{p}(i)\right\|^2\left\|\mathbf{y}\right\|^2\mathbf{e}_1\mathbf{e}_1^H\right)$ $\mathbf{D} = \mathbf{I}$, we finally obtain

$$\begin{aligned}\varepsilon(i) = \mathbf{D}(i)\mathbf{H}(i)[\varepsilon(i-1) + \beta(\varepsilon(i-1)\mathbf{\Omega}(i) + \mathbf{\Omega}(i)\varepsilon(i-1))] \\ \mathbf{H}(i)\mathbf{D}(i)\end{aligned}$$

which leads to

$$vec(\varepsilon(i)) = \mathbf{M}(i)vec(\varepsilon(i-1)) + \mathbf{b}(i). \quad (8)$$

With

$$\mathbf{M}(i) = \mathbf{DH}(i)\otimes\mathbf{DH}(i)(\mathbf{I}_{p^2} + \beta(\mathbf{\Omega}(i)\otimes\mathbf{I}_p + \mathbf{I}_p\otimes\mathbf{\Omega}(i)))$$

$\mathbf{b}(i)$ represents the instantaneous error due of the normalization step, i.e. $\mathbf{D}(i)\left(\mathbf{I} + \beta^2\left\|\mathbf{p}(i)\right\|^2\left\|\mathbf{y}\right\|^2\mathbf{e}_1\mathbf{e}_1^H\right)\mathbf{D}(i) = \mathbf{I} + \mathbf{B}(i)$, where $\mathbf{b}(i) = vec(\mathbf{B}(i))$ and $vec(.)$ represents the column vectorization operator.
Clearly, matrix $\mathbf{M}(i)$ has eigenvalues larger than 1 (this can be seen using a first order approximation with respect to β). Hence, the deviation from orthonormality does increase at each iteration.
Remark: When analyzing the principal subspace, we obtain the same form as in (8) but with an opposite sign of β. In this case all eigenvalues of $\mathbf{M}(i)$ are smaller than 1 and hence FOOja algorithm is absolutely stable for PSA.

4.2. Numerical stability of FDPM

Using a similar analysis as for FOOja algorithm, we get for the FDPM:

$$vec(\varepsilon(i)) = \ \mathbf{Q}(i)vec(\varepsilon(i-1)) + \mathbf{b}(i) \quad (9)$$

where $\mathbf{Q}(i) = \mathbf{D}(i)\mathbf{H}(i)\otimes\mathbf{D}(i)\mathbf{H}(i)$ and
$\mathbf{D}(i) = diag(\frac{1}{\sqrt{1-(2\beta-\beta^2)\frac{\|\mathbf{y}(i)\|^2}{\|\mathbf{r}(i)\|^2}}}, 1, \ldots, 1)$. We can see clearly that $\frac{1}{\sqrt{1-(2\beta-\beta^2)\frac{\|\mathbf{y}(i)\|^2}{\|\mathbf{r}(i)\|^2}}}$ is larger than 1. Therefore we demonstrate that this method is not stable at least when normalizing just the first column of $\mathbf{W}(i)$.

4.3. Numerical stability of OOjaH

Using equations of Table 1, we form $\mathbf{W}^H(i)\mathbf{W}(i)$ and after some manipulations we get:

$$\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{W}^H(i-1)\mathbf{W}(i-1) + 4\left(\|\mathbf{u}(i)\|^2 - 1\right)\mathbf{v}(i)\mathbf{v}^H(i)$$

The deviation from orthonormality can be written:

$$\varepsilon(i) = \varepsilon(i-1) + 4\left(\|\mathbf{u}(i)\|^2 - 1\right)\mathbf{v}(i)\mathbf{v}^H(i) \quad (10)$$

We remark that the principal source causing the deviation from orthonormality is the instantaneous error of normalization of $\bar{\mathbf{p}}(i)$ i.e. $\mathbf{u}(i) = \frac{\bar{\mathbf{p}}(i)}{\|\bar{\mathbf{p}}(i)\|}$. The simulations (figure 1 and 2) illustrate the effect of this instantaneous numerical error on the deviation from orthonormality.

5. SIMULATIONS

To assess the performance of our algorithms, we calculate the ensemble average of the performance factors $\rho(i) = \frac{1}{r_0}\sum_{r=1}^{r_0}\frac{tr(\mathbf{W}^H(i)\mathbf{E}_1\mathbf{E}_1^H\mathbf{W}(i))}{tr(\mathbf{W}^H(i)\mathbf{E_2}\mathbf{E}_2^H\mathbf{W}(i))}$ and $\eta(i) = \frac{1}{r_0}\sum_{r=1}^{r_0}\|\mathbf{W}_r^H(i)$ $\mathbf{W}_r(i) - \mathbf{I}\|_F^2$ where the number of algorithm runs is $r_0 = 50$, r indicates that the associated variable depends on the particular run. $\|.\|_F$ denotes the Frobenius norm, and $\mathbf{E_2}$ (resp. $\mathbf{E_1}$) is the matrix of the p (resp. $n-p$) minor (resp. principal) eigenvectors. The first performance index ρ measures the averaged estimation accuracy of the minor subspace while the second performance index η measures the orthogonality of the weight matrix (for PSA, we replace in ρ, $\mathbf{E}_1$ by $\mathbf{E}_2$ and vis versa). In the simulation experiment, we have considered an iid sequence of n-dimensional (with $n = 10$) random vectors $\mathbf{x}(i)$. 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. The minor subspace dimension is equal to $p = 4$. The used step size is $\beta = 0.7$ for the step-size normalized version of FDPM, FOOja and OOjaH (with $\gamma = 0.65$, [8]).
In Fig.1, we compare the different algorithms (MS-OOjaH, MS-FOOja and MS-FDPM). As we can see, MS-FOOja behaves slightly better than MS-FDPM and MS-OOjaH.
In Fig.2 we introduce a complete deviation from orthonormality at the 2000 iteration. We observe that OOjaH diverges from orthonormality and becomes instable while FOOja

and FDPM remain stable and converge quickly to orthonormality thanks to the orthogonalization step at each iteration. In figure 3, we show by simulation the divergence from orthogonality (i.e. $\mathbf{W}(i)$ becomes singular) that we have proved when analyzing the numerical stability of the MS-FOOja and MS-FDPM algorithms if the first column vector of $\mathbf{W}(i)$ is only normalized. However, for PSA, these algorithms are absolutely stable as shown by the stability analysis and illustrated by Fig. 4.

6. REFERENCES

[1] M. Moonen, P. Van Dooren, and J. Vandewalle, "Updating singular value decompositions: A parallel implementations," *in Proc. SPIE Adv. Alg. Archi. Sig. Proc.*, pp. 80–91, 1989.

[2] X. G. Doukopoulos and G. V. Moustakides, "The fast data projection method for stable subspace tracking," *EUSIPCO*, september 2005.

[3] T.P. Krasulina, "Method of stochastic aprrroximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices," *Automat. Remote Contr.*, vol. 2, pp. 215–221, 1970.

[4] T. Chen, S.-I Amari, and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, pp. 385–390, Aug. 1998.

[5] E.Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, Dec. 1992.

[6] K. Abed-Meraim, S. Attallah, A. Chkeif, and Y. Hua, "Orthogonal oja algorithm," *IEEE SP Letters*, pp. 116–119, 2000.

[7] G.H. Golub and C.F. Van loan, "Matrix computations," *Baltimore, MD: Johns Hopkings Univ. Press*, pp. 205–213, 1996.

[8] S. Attallah and K. Abed-Meraim, "Subspace estimation and tracking using enhanced versions of oja's algorithm," *IEEE Sig. Proc. advances in Wire. Com.*, pp. 218–220, March 2001.

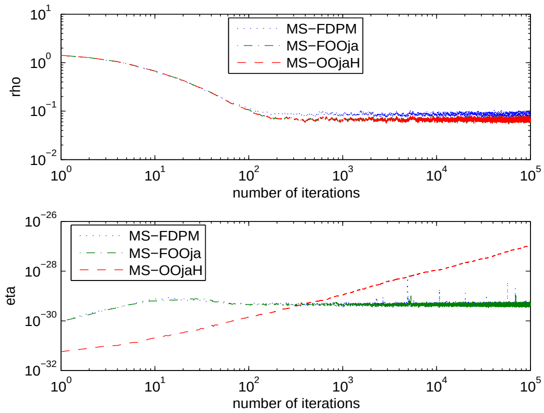

Fig. 1. Performance of MS-FOOja, MS-OOjaH and MS-FDPM

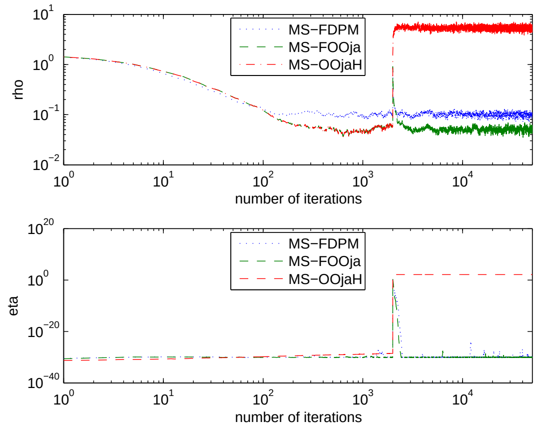

Fig. 2. Complete deviation from orthonormality

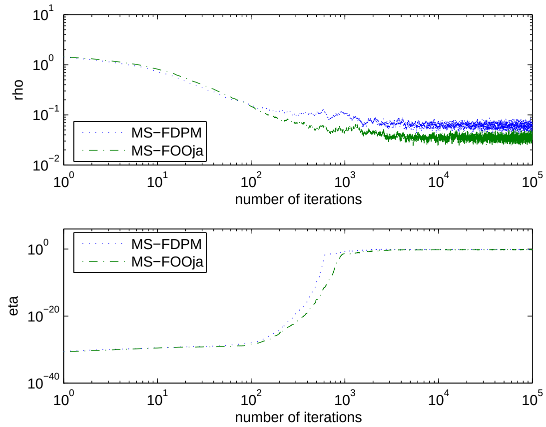

Fig. 3. Divergence of FOOja and FDPM when only normalizing the first column vector of $\mathbf{W}(i)$

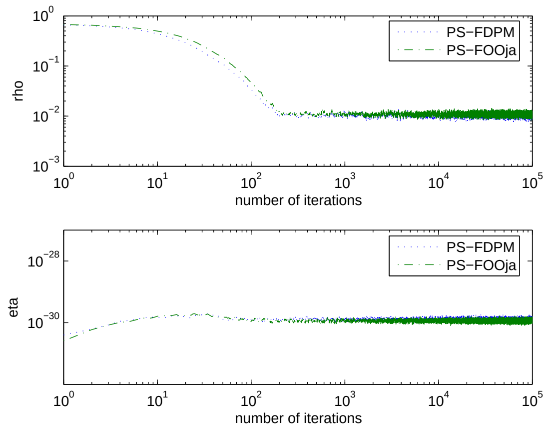

Fig. 4. Absolute Stability of PS-FOOja and PS-FDPM