MINOR SUBSPACE TRACKING USING MNS TECHNIQUE

Messaoud Thameri[⋆] *Karim Abed-Meraim*[⋆] *Adel Belouchrani*[⋆⋆]

[⋆] Telecom ParisTech, 46 rue Barrault, 75634 Paris Cedex 13, France
[⋆⋆]Ecole Nationale Polytechnique, 10 Avenue Hassen Badi, 16200 Algiers, Algeria
thameri@telecom-paristech.fr, karim.abed@telecom-paristech.fr, adel.belouchrani@enp.edu.dz

ABSTRACT

This paper introduces new minor (noise) subspace tracking (MST) algorithms based on the minimum noise subspace (MNS) technique. The latter has been introduced as a computationally efficient subspace method for blind system identification. We exploit here the principle of the MNS, to derive the most efficient algorithms for MST. The proposed method joins the advantages of low complexity and fast convergence rate. Moreover, this method is highly parallelizable and hence its computational cost can be easily reduced to a very low level when parallel architectures are available. Different implementations are proposed for different contexts and they are compared via numerical simulations.

***Index Terms*—** MNS, Minor subspace, Fast adaptive algorithm

1. INTRODUCTION

Fast estimation and tracking of minor (noise) subspaces is an important problem in many applications related to the fields of telecommunication and array signal processing [1, 2].

This problem is known for many decades and several solutions exist in the literature including the Oja (gradient) like techniques of linear complexity [3, 4, 5] (i.e, of complexity $O(nm)$ flops per iteration where n is the size of the observation vector and m is the rank of the desired minor subspace) and the power like techniques (YAST, PAST) of quadratic complexity $O(n^2)$ [6, 7].

In this paper, we consider the typical situation where an array of sensors receives signals from p impinging sources corrupted by additive white noise so that the observation vector writes as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \tag{1}$$

where $\mathbf{x}(t)$ is of size $n \times 1$, $\mathbf{s}(t)$ is the $p \times 1$ sources vector, $\mathbf{n}(t)$ is the observation noise of covariance $\mathbf{E}[\mathbf{n}(t)\mathbf{n}^H(t)] = \sigma^2\mathbf{I}$ and $\mathbf{A}$ is the $n \times p$ full column rank array matrix.

We are interested in the adaptive estimation and tracking of the noise subspace corresponding to the $m = n - p$ dimensional eigensubspace associated to σ^2, the least eigenvalue of the observed data covariance matrix $\mathbf{R} = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma^2\mathbf{I}$.

For that, we adopted the principle of MNS [1] [9] based on which new tracking algorithms are proposed. The latter are characterized by their reduced computational cost which is (for some of the proposed implementations) as low as $O(pm)$ flops per iteration [2]. Besides their advantage in term of numerical complexity, the proposed algorithms have, in most of the considered cases, the fast convergence rate and good estimation accuracy as compared to existing methods.

Finally, another non negligeable advantage of our method is its parallelizable structure inherent to the MNS approach, a feature that takes more and more importance due to the growing interest in the distributed estimation for sensors network [10] as well as the growing demand for the use of parallel architectures and multi-core computing facilities [11].

[1]This is a method introduced in the context of blind identification of MIMO systems [8] in order to reduce the computational cost of the subspace based method.

[2]To our best knowledge, this is the lowest computational cost ever proposed for solving this MST problem.

2. MNS FOR NOISE SUBSPACE EXTRACTION

The MNS method has been introduced in [8] as a fast subspace technique for the blind identification of MIMO (Multiple Input Multiple Output) systems. Latter on, this concept was shown to be applicable to other array processing problems where the observation vector is corrupted by additive white noise [9].

The MNS consists in using a set of subsystems (sub-arrays) of minimum noise subspace rank[3], then extracting the desired noise vectors from the least eigenvectors of the covariance matrices of the considered subsystems. The selection of the subsystems is achieved according to the PCS (Properly Connected Sequence) concept as summarized below.

2.1. PCS concept

For an n dimensional system, a subsystem of size $p+1$ is represented by a $(p+1)$-tuple $t = (m_{i_1}, \cdots, m_{i_{p+1}})$ regrouping the system sensors used in this system. Then, the PCS is defined as follows [9].

Definition: A sequence of $n-p$ tuples is said to be properly connected if each tuple in the sequence consists of p members shared by its preceding tuples and another member not shared by its preceding tuples. A properly connected sequence (PCS) is denoted by $S(p,n) = (t_1, t_2, \cdots, t_{n-p})$ where

$$\begin{aligned} &t_i = (m_{i_1}, \cdots, m_{i_p}, m_{i_{p+1}}), 1 \le i \le n-p \\ &\{m_{i_1}, \cdots, m_{i_p}\} \subset \{m_{j_k} \mid j < i, 1 \le k \le p+1\} \\ &m_{i_{p+1}} \notin \{m_{j_k} \mid \forall j < i, 1 \le k \le p+1\} \end{aligned}$$

2.2. Computation of the MNS

The data covariance matrix of the i^{th} subsystem writes as:

$$\mathbf{R}^i = \mathbf{E}[\mathbf{x}^i(t)\mathbf{x}^i(t)^H] = \mathbf{A}^i\mathbf{R}_s\mathbf{A}^{iH} + \sigma^2\mathbf{I} \tag{2}$$

where $\mathbf{x}^i(t) = [x_{i_1}, \cdots, x_{i_{p+1}}]^T$ is the observed vector of the sub-system given by the outputs member of the i^{th} tuple $t_i = (m_{i_1}, \cdots, m_{i_p}, m_{i_{p+1}})$ and $\mathbf{A}^i$ is its corresponding response

[3]For each subsystem, the corresponding noise subspace is of rank 1.

matrix. Let $\tilde{\mathbf{v}}^i$ be the least eigenvector of $\mathbf{R}^i$. A set of vectors $\{\mathbf{v}^i\}_{1\leq i\leq m}$ forming a basis of the noise subspace are constructed in the following way: for $j = 1, \cdots, n$,

$$v^i(j) = \begin{cases} 0 & \text{if the } j^{th} \text{ output of the system does} \\ & \text{not belong to the } i^{th} \text{ tuple.} \\ \tilde{v}^i(j') & \text{if the } j^{th} \text{ output of the system} \\ & \text{is the } j'^{th} \text{ member of the } i^{th} \text{ tuple.} \end{cases}$$

For simplicity and also to reduce the computational cost, the PCS considered is this paper is the one given by:

$$t_i = (m_1, m_2, \cdots, m_p, m_{p+i}), \quad i = 1, 2, \cdots, n-p$$

In the sequel, we assume that the first p row vectors of matrix $\mathbf{A}$ form a full rank $p \times p$ matrix.

3. MNS-BASED MTS ALGORITHMS

In this section, we introduce new adaptive algorithms for the MST using the MNS concept. These algorithms have different performances depending on the used technique to estimate the least eigenvector of each of the covariance matrices $\mathbf{R}^i$, $i = 1, \cdots, m$ and on the considered context.

3.1. MNS-OOja

At first, we consider the simplest (and hence cheapest) algorithm for the least eigenvector extraction, namely the OOja [3] which is a gradient type method of linear complexity. The estimate of the least eigenvector $\mathbf{v}^i(t)$ is updated at time instant t as

$$y^i(t) = \mathbf{v}^i(t-1)^H \mathbf{x}^i(t) \quad (3)$$

$$\mathbf{v}^i(t) = (1 + \alpha|y^i(t)|^2)\mathbf{v}^i(t-1) - \alpha y^i(t)^* \mathbf{x}^i(t) \quad (4)$$

$$\mathbf{v}^i(t) := \mathbf{v}^i(t)/\|\mathbf{v}^i(t)\| \quad (5)$$

where $\alpha > 0$ is a learning parameter. This technique leads to an overall computational complexity of order $2pm + O(m)$ but suffers from low convergence rate (or even divergence) as compared to other existing methods especially when the size of the subsystem is large. To improve the convergence performance, we propose next to replace the OOja by YAST [6], a more expensive but more efficient tracking algorithm.

3.2. MNS-YAST-MS

In this implementation, we use YAST for minor subspace (MS) to estimate the least eigenvector of $\mathbf{R}^i(t)$ given by

$$\mathbf{R}^i(t) = \beta \mathbf{R}^i(t-1) + (1-\beta)\mathbf{x}^i(t)\mathbf{x}^i(t)^H \quad (6)$$

where β is the forgetting factor $0 < \beta < 1$. The main steps of YAST to extract vector $\mathbf{v}^i$ are as follows:

1. Update $\mathbf{R}^i(t)$ as in (6).
2. Form $\mathbf{V}^i(t) = \left[\mathbf{v}^i(t-1), \overline{\mathbf{x}}^i(t)\right]$ where $\overline{\mathbf{x}}^i(t) = (\mathbf{x}^i(t) - y^i(t)\mathbf{v}^i(t-1))/\|\mathbf{x}^i(t) - y^i(t)\mathbf{v}^i(t-1)\|$.
3. Compute the 2×2 matrix $\mathbf{C}^i(t) = \mathbf{V}^i(t)^H \mathbf{R}^i(t) \mathbf{V}^i(t)$.
4. Compute the least eigenvector $\mathbf{u}^i(t)$ of $\mathbf{C}^i(t)$.
5. Compute $\mathbf{v}^i(t) = \mathbf{V}^i(t)\mathbf{u}^i(t)$

The main cost (of order $O((p+1)^2)$ comes from steps 1 and 3 related to the updating of matrix $\mathbf{R}^i(t)$ and the matrix vector products, $\mathbf{R}^i(t)\mathbf{v}^i(t-1)$ and $\mathbf{R}^i(t)\mathbf{x}^i(t)$. As shown in [6], the first matrix vector product can be simplified using the projection approximation and hence its cost is reduced to $O(p)$. In our case, one can reduce the computational cost of the two other terms by taking advantage of the fact that all covariance matrices $\mathbf{R}^i$ have a common $p \times p$ bloc:

$$\mathbf{R}^i = \begin{bmatrix} \mathbf{M} & \mathbf{s}^i \\ \mathbf{s}^{iH} & d^i \end{bmatrix} \quad (7)$$

where $\mathbf{M}$ is a common $(p \times p)$ matrix, $\mathbf{s}^i$ is a $p \times 1$ vector and d^i is a scalar. As we can see, the updating of $\mathbf{R}^i$ requires the updating of $\mathbf{M}$ which is done only once and costs p^2 flops as well as the updating of vectors $\mathbf{s}^i$ and scalars d^i for $i = 1, \cdots, m$ which costs $O(pm)$ flops. Similarly, the numerical cost of the products $\mathbf{R}^i(t)\mathbf{x}^i(t)$ for $i = 1, \cdots, m$ can be reduced by observing that

$$\mathbf{R}^i\mathbf{x}^i = \begin{bmatrix} \mathbf{M}\mathbf{x}_{1:p} + \mathbf{s}^i x_{p+i} \\ \mathbf{s}^{iH}\mathbf{x}_{1:p} + d^i x_{p+i} \end{bmatrix} \quad (8)$$

which shows that the term $\mathbf{M}\mathbf{x}_{1:p}$ can be computed only once. Consequently, the overall cost of this algorithm is $2p^2 + O(mp)$.

In our simulations, we observed that this algorithm achieves its best performance when $p < m$. For the case $p \geq m$, the algorithm's convergence can be significantly improved by using the inverse matrix $\mathbf{R}^i(t)^{-1}$ instead of $\mathbf{R}^i(t)$ for the computation of vector $\mathbf{v}^i$.

3.3. MNS-YAST-PS

Here, we propose to use YAST to estimate the Principal Subspace[4] (PS) of $\mathbf{R}^i(t)^{-1}$ which is updated as follows using Schur's inversion lemma [12]

$$\mathbf{R}^i(t)^{-1} = \frac{1}{\beta}(\mathbf{R}^i(t-1)^{-1} - f^i \mathbf{x}_r^i(t)\mathbf{x}_r^i(t)^H) \quad (9)$$

where $\mathbf{x}_r^i(t) = \mathbf{R}^i(t-1)^{-1}\mathbf{x}^i(t)$ and $f^i = \frac{1-\beta}{\beta+(1-\beta)\mathbf{x}^i(t)^H\mathbf{x}_r^i(t)}$. This algorithm has the same steps as MNS-YAST-MS except that it uses $\mathbf{R}^i(t)^{-1}$ instead of $\mathbf{R}^i(t)$, $\mathbf{x}_r^i(t)$ instead of $\mathbf{x}^i(t)$ and in step 4 we use the principal (instead of least) eigenvector $\mathbf{u}^i(t)$ of $\mathbf{C}^i(t)$.

A direct computation would cost $O(p^2 m)$ flops per iteration. However, by considering the particular structure of matrices $\mathbf{R}^i$, one can reduce this cost to $3p^2 + O(mp)$ as shown below.

First, by using the bloc matrix inversion lemma, one can write:

$$\begin{bmatrix} \mathbf{M} & \mathbf{s}^i \\ \mathbf{s}^{iH} & d^i \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} + \mathbf{M}^{-1}\mathbf{s}^i\, g^i\, \mathbf{s}^{iH}\mathbf{M}^{-1} & -\mathbf{M}^{-1}\mathbf{s}^i\, g^i \\ -g^i\, \mathbf{s}^{iH}\mathbf{M}^{-1} & g^i \end{bmatrix}$$

where $g^i = (d^i - \mathbf{s}^{iH}\mathbf{M}^{-1}\mathbf{s}^i)^{-1}$. Therefore, the fast inversion of $\mathbf{R}^i(t)$ consists in the fast inversion of matrix $\mathbf{M}(t)$ using the inversion lemma shown in (9), i.e.

$$\mathbf{M}^{-1}(t) = \frac{1}{\beta}(\mathbf{M}^{-1}(t-1) - f\mathbf{z}(t)\mathbf{z}^H(t)) \quad (10)$$

where $\mathbf{z}(t) = \mathbf{M}^{-1}(t-1)\mathbf{x}_{1:p}(t)$ and $f = \frac{1-\beta}{\beta+(1-\beta)\mathbf{x}_{1:p}^H(t)\mathbf{z}(t)}$, and the fast computation of matrix vector products $\mathbf{M}^{-1}\mathbf{s}^i$ according to

[4] The least eigenvector of $\mathbf{R}^i(t)$, i.e. vector $\mathbf{v}^i$, is the principal eigenvector of its inverse matrix $\mathbf{R}^i(t)^{-1}$.

the decomposition:

$$\mathbf{M}^{-1}(t)\mathbf{s}^i(t) = \frac{1}{\beta}(\mathbf{M}^{-1}(t-1) - f\mathbf{z}(t)\mathbf{z}^H(t)) \times \quad (11)$$

$$(\beta\mathbf{s}^i(t-1) + (1-\beta)\mathbf{x}_{1:p}(t)x^*_{p+i}(t)) \quad (12)$$

$$= \mathbf{M}^{-1}(t-1)\mathbf{s}^i(t-1) + \lambda\mathbf{z}(t) \quad (13)$$

where $\lambda = \frac{1-\beta}{\beta}x^*_{p+i}(t)(1 - f\mathbf{z}^H(t)\mathbf{z}(t)) - f\mathbf{z}^H(t)\mathbf{s}^i(t-1)$.

In addition to the previous calculations, in the fast implementation of YAST, one needs to compute the matrix vector products $\mathbf{R}^i(t-1)^{-1}\mathbf{x}^i_r(t)$. Using the bloc inversion form of $\mathbf{R}^i(t-1)^{-1}$, as well as equation (10) and the updating equation of $\mathbf{s}^i(t)$, one can easily show that the above products can be computed in $p^2 + O(mp)$ flops per iteration.

3.4. MNS-Hybrid

In the case where a parallel computer architecture is available or if $p \ll n$, we propose here a last implementation version that costs $O(mp^2)$ (or $O(p^2)$ if m processors are used in parallel) that has the advantage of faster convergence as compared to the previous algorithms. It is based on the use of the modified version of the Hybrid method in [13] which is an appropriate combination between power method and gradient based method. As in MNS-YAST-PS, the Hybrid method is used to estimate the largest eigenvector of $\mathbf{R}^i(t)^{-1}$ in a very similar way to YAST-PS except that it does not use the projection approximation. The main steps of Hybrid method are

1. Update $\mathbf{R}^i(t)^{-1}$ as in (9).
2. Compute $\mathbf{w}^i = \mathbf{R}^i(t)^{-1}\mathbf{v}^i(t-1)$
3. Form $\mathbf{V}^i = \left[\mathbf{v}^i(t-1), \overline{\mathbf{w}}^i\right]$ where $\overline{\mathbf{w}}^i = (\mathbf{w}^i - \mathbf{v}^i(t-1)^H\mathbf{w}^i\mathbf{v}^i(t-1))/\|\mathbf{w}^i - \mathbf{v}^i(t-1)^H\mathbf{w}^i\mathbf{v}^i(t-1)\|$.
4. Compute $\mathbf{B}^i = \mathbf{V}^{iH}\mathbf{R}^i(t)^{-1}\mathbf{V}^i$
5. Compute the principal eigenvector $\mathbf{e}^i$ of $\mathbf{B}^i$
6. The desired eigenvector is given by $\mathbf{v}^i(t) = \mathbf{V}^i\mathbf{e}^i$.

4. DISCUSSION

We provide here comments that highlight the advantages as well as the drawbacks of the proposed MNS-based method.

Numerical complexity: As mentioned earlier, the main advantage of the proposed algorithms is their low computational cost that ranges from $O(mp)$ for the MNS-OOja to $O(p^2m)$ for the MNS-Hybrid. For $p \ll n$ this complexity is much lower than the linear complexity $O(mn)$ of gradient like techniques. This cost reduction is made possible thanks to the specific structure of the covariance matrix (i.e all noise subspace vectors are associated to the same eigenvalue σ^2). Note that this structure has already been exploited in [14] for the derivation of the fast (batch) subspace method of complexity $O(n^2p)$.

Parallelizable structure: The noise vectors are computed in the MNS method from different subsystems. Hence, they might be computed in a parallel scheme if one can afford one computing unit per subsystem, in which case the computational cost reduces to $O(p)$ or $O(p^2)$ flops per iteration. Note that the parallel computing is a growing interest issue boosted by the fast development of parallel computer architectures [11].

Convergence rate and estimation accuracy: Surprisingly, the convergence rate as well as the noise subspace estimation accuracy of the MNS based MST algorithms are relatively comparable (if not better) than those of the recently proposed YAST algorithm. Indeed, the eventual performance loss due to the use of a small number of observation outputs (i.e. we use $p+1$ system outputs instead of n) for the computation of the noise vectors is compensated for by the side information we exploit related to the specific covariance matrix structure. In particular, the simulation results show the good behaviour or our algorithms in the case of large dimensional systems, i.e. systems of size $n \gg 1$.

Orthonormalization: A drawback of the proposed method resides in the fact that the computed noise subspace basis is not orthonormal. An exact orthonormalization using Gram-Schmidt technique would cost $O(p^2m)$ flops per iteration. In general, for most applications, the exact orthonormality is not a strict requirement, in which case it would be preferable to use approximate orthonormalization techniques in order to preserve the low complexity of the proposed algorithms.

Ill conditioned subsystem matrices: By using only $p+1$ outputs per subsystem, the risk to have ill conditioned covariance matrices increases leading to ill convergence and poor estimation accuracy. A possible solution to this problem, would be to generalize the MNS approach and use $p+d$ outputs per subsystem where $d \geq 1$ is a chosen integer parameter that controls the trade off between the computational cost and the algorithm's convergence performance. This generalization will be the focus of future works.

5. SIMULATION

The performance of the MNS-based MTS algorithms are assessed in this section. The proposed algorithms are compared to YAST-PS applied to the inverse covariance matrix $\mathbf{R}^{-1}(t)$ for the estimation and tracking of the noise subspace[5] as well as the MNS-SVD algorithm where the Singular Value Decomposition (SVD) is used to estimate the least eigenvector of $\mathbf{R}^i(t)$. These two algorithms have been chosen as references because of their high performance level. The performance index is defined as [6]

$$\eta(t) = \frac{1}{p_0}\sum_{k=1}^{p_0}\frac{\text{trace}(\mathbf{\Pi}_A\mathbf{\Pi}_{W_k}(t))}{\text{trace}((\mathbf{I}-\mathbf{\Pi}_A)\mathbf{\Pi}_{W_k}(t))}$$

where $p_0 = 50$ is the number of Monte-Carlo runs, $\mathbf{W}_k(t)$ is the estimated noise subspace matrix at the k-th run and iteration t and $\mathbf{\Pi}_A$ (resp. $\mathbf{\Pi}_{W_k}(t)$) is the orthogonal projection matrix on the range space of $\mathbf{A}$ (resp. on the range space of $\mathbf{W}_k(t)$).

We have chosen two different contexts for a relatively large dimensional system to show the effect of p on the proposed algorithms. Context 1 ($n = 20$, $p = 3$) represented by Fig.1 and context 2 ($n = 20$, $p = 13$) represented by Fig.2. For both contexts, matrix $\mathbf{A}$ is generated following the directional matrix model of linear uniform arrays with p randomly selected angle of arrivals, the sources are independent i.i.d. gaussian processes of unit power and the noise is spatially white of power $\sigma^2 = 0.02$.

Simulation results of context 1 show that the MNS-OOja algorithm has the lowest performance accuracy and convergence speed

[5] Indeed YAST-PS with $\mathbf{R}^{-1}(t)$ has faster convergence and better estimation accuracy than YAST-MS with $\mathbf{R}(t)$.

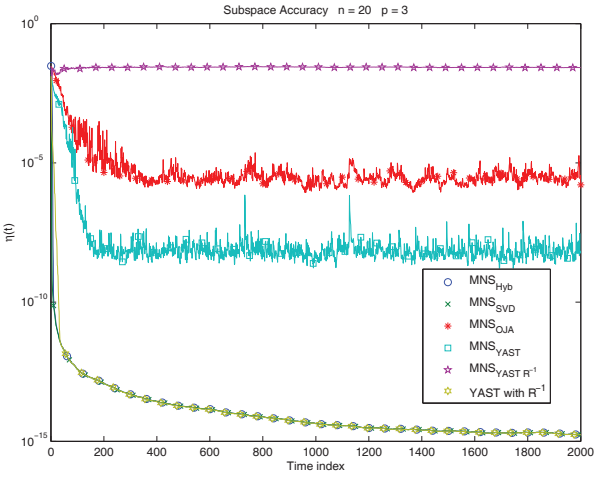

Fig. 1. Comparison of proposed algorithms for n=20 and p=3

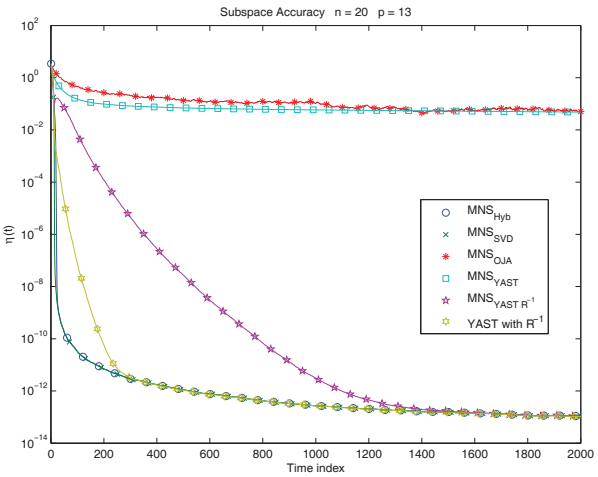

Fig. 2. Comparison of proposed algorithms for n=20 and p=13

followed by the MNS-YAST-MS algorithm then the MNS-Hybrid. The latter algorithm reaches the same performance as MNS-SVD and the global YAST-PS. Note also, that in this context the MNS-YAST-PS algorithm diverged for most of the simulation runs.

When p becomes large (i.e. context 2), MNS-YAST-MS and MNS-OOja algorithms suffer from low convergence rate or even divergence. As we can see from Fig.2, the MNS-YAST-PS shows good performance in this context but again, the MNS-Hybrid presents the best accuracy and convergence rate performance as compared to all other algorithms considered in this work.

6. CONCLUSION

To our best knowledge, this paper proposes the most efficient noise subspace tracking algorithms for the case where the observation vector lives in a p dimensional (signal) subspace and is corrupted by additive white noise. Our method is inspired from the MNS, a technique introduced in the context of blind system identification. The proposed algorithms have low numerical complexity, parallelizable structures and good convergence and estimation performance. Different implementations which trade off between complexity (varying from $O(pm)$ to $O(p^2m)$) and convergence performance are considered. The conducted simulation experiments confirm the good behavior of our method as compared to YAST one of the most effective tracking algorithms in the literature.

7. REFERENCES

[1] X. G. Doukopoulos, "Power techniques for blind channel identification in wireless communication systems," *PhD. thesis, INRIA, French*, 2004.

[2] J. P. Delmas, *Chapter 4. Subspace Tracking for Signal Processing*, John Wiley and Sons, Inc., 2010.

[3] K. Abed-Meraim, S. Attallah, A. Chkeif, Y. Hua, "Orthogonal Oja algorithm ," *Signal Processing Letters, IEEE*, May 2000.

[4] S. Bartelmaos, K. Abed-Meraim , "Fast adaptive algorithms for minor component analysis using Householder transformation," *DSP journal*, accepted 2011.

[5] S.C.Douglas, S.-Y Kung, S.-I Samari, "A self-stabilized minor subspace rule," *Sig. Proc. Letters*, 1998.

[6] R. Badeau, G. Richard, B. David, "Fast and Stable YAST Algorithm for Principal and Minor Subspace Tracking," *IEEE Transaction on Signal Proceeding*, 2008.

[7] H. Sakai, K. Shmizu, "A new adaptative algorithm for minor component analysis," *Signal Processing*, 1998.

[8] K. Abed-Meraim and Y. Hua, "Blind identification of multi-input multi-output system using minimum noise subspace," *IEEE Tr. on SP*, 1997.

[9] K. Abed-Meraim, Y. Hua, A. Belouchrani, "Minimum noise subspace: concepts and applications ," *ICICS*, 1997.

[10] K.-U Lei, Yao-Win Hong,; C.-Y. Chi , "Distributed estimation in wireless sensor networks with channel-aware slotted aloha," *SPAWC*, 2007.

[11] Jakub Kurzak, David A. Bader, Jack Dongarra , *Scientific Computing with Multicore and Accelerators (Chapman and Hall CRC Computational Science)*, Cambridge University Press; 3 edition, 2011.

[12] 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.

[13] P. Comon, G.H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proceedings of the IEEE*, 1990.

[14] Xu Guanghan, T. Kailath, "Fast subspace decomposition," *IEEE Transaction on Signal Proceeding*, 1994.