



**HAL**  
open science

# Array Processing Using Generalized Minimum Noise Subspace

Viet-Dung Nguyen, Karim Abed-Meraim, Nguyen Linh-Trung, Rodolphe Weber

► **To cite this version:**

Viet-Dung Nguyen, Karim Abed-Meraim, Nguyen Linh-Trung, Rodolphe Weber. Array Processing Using Generalized Minimum Noise Subspace. 2016. hal-01295030v1

**HAL Id: hal-01295030**

**<https://inria.hal.science/hal-01295030v1>**

Preprint submitted on 30 Mar 2016 (v1), last revised 8 Jun 2017 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Array Processing Using Generalized Minimum Noise Subspace

Viet-Dung Nguyen, Karim Abed-Meraim, Nguyen Linh-Trung, and Rodolphe Weber

## Abstract

In this paper, we present a new array processing scheme which allows fast and parallel computation of the signal (principal) and noise (minor) subspaces. The proposed method, named Generalized Minimum Noise Subspace (GMNS), is a generalization of the minimum noise subspace concept, introduced first for blind channel identification, subject to an availability of a fixed number of parallel computational units. In addition, we introduce different batch and adaptive algorithms for fast and parallel computation of the noise as well as the signal subspaces. The GMNS cost and its related estimation accuracy are investigated by theoretical and numerical analyses. The latter shows that the GMNS represents an excellent trade-off between the computational gain and estimation loss as compared to the standard subspace methods.

## Index Terms

Batch and adaptive algorithms, principal and minor subspace, MNS, GMNS, PCA, MCA, parallel computing.

## I. INTRODUCTION

Principal subspace analysis (PSA) and minor subspace analysis (MSA) play important roles in many practical signal processing applications such as high resolution parameter estimation [1], blind source separation [2], and radio frequency interference mitigation [3]. Important problems

V. D. Nguyen, K. Abed-Meraim, and R. Weber are with PRISME laboratory, University of Orléans, 12 rue de Blois BP 6744, Orléans, France. Emails: {viet-dung.nguyen, karim.abed-meraim, rodolphe.weber}@univ-orleans.fr.

R. Weber is also with Station de radioastronomie de Nancay, Observatoire de Paris, CNRS/INSU, F-18330 Nancay, France

N. Linh-Trung is with the University of Engineering and Technology, Vietnam National University Hanoi, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam. E-mail: linhtrung@vnu.edu.vn.

closely related to PSA and MSA are principal component analysis (PCA) and minor component analysis (MCA); they usually require eigen-subspaces of the considered data covariance matrix [4].

For batch systems, standard subspace techniques based on singular value decomposition (SVD) or eigenvalue decomposition (EVD) are often applied. Although these techniques have high performance advantages, they face a high computational cost, generally of  $O(n^3)$  operations, where  $n$  is the dimension of the observation vector or the number of sensors. This cost causes a serious handicap for large-dimensional systems such as large sensor networks [5], massive MIMO systems [6], or large antenna arrays like the square kilometer array used in radio astronomy [7]. In addition, SVD-like methods are not suitable for adaptive subspace tracking because they require repeated decompositions. In such a case, the use of distributed algorithms [8], parallel computation schemes [3], and fast adaptive techniques [9]–[11] becomes of high interest and generally leads to large gains in terms of computational cost and memory requirements.

Supposed that we have at hand a parallel computing architecture with  $K$  computational units. The question of interest in this paper is how can we exploit this architecture to reduce the numerical cost for extracting the minor or principal subspaces? An interesting subspace method called minimum noise subspace (MNS) was proposed in [12]. This method estimates the noise subspace via a set of noise vectors (basis of the noise subspace) that can be computed *in parallel* from a set of tuples of system outputs. Because of avoiding the EVD computation at large scale, the MNS is much more efficient in computation than standard subspace methods. The MNS has been applied to blind system identification [12], source localization [13], array calibration [13], multichannel blind image deconvolution [14], and adaptive subspace tracking [15]. However, the number of parallel computing units based on which the MNS is implemented is the same as the dimension of the noise subspace, and is in general not equal to the actually available number of parallel units ( $K$ ) of the computing architecture in use. In addition, the performance of the MNS is degraded if the number of outputs is very large compared to that of the inputs.

In this paper, we follow the MNS method but generalize it in such a way that we can handle the MNS computation with a given  $K$  number of parallel computing units. The main contributions are summarized as follows.

- 1) Via the introduction of the concept of a generalized properly connected sequence (GPCS), we propose a generalized minimum noise subspace (GMNS) method for efficient minor subspace analysis (MSA) given a fixed number of parallel computational units. It is noted

that in our conference contribution [16] we have disseminated partial results on GMNS and in this paper we provide detailed proofs and extensive experiments.

- 2) We then propose a procedure to estimate principal or minor eigenpairs from the corresponding principal and minor subspaces (i.e., solving the PCA and MCA problems) by solving a joint diagonalization problem.
- 3) We develop efficient adaptive versions of the proposed algorithms for both principal subspace tracking (PST) and minor subspace tracking (MST) by integrating several available subspace tracking procedures in our parallel framework. As will be shown, our algorithms have advantages of low complexity and fast convergence rate. Note that some adaptive algorithms for MNS have already been proposed in [17] but they are limited to least minor subspace analysis.
- 4) We further propose efficient adaptive algorithms for principal eigenvector tracking from the corresponding PST. The performance of our algorithms is nearly identical to the standard SVD-based algorithm with less cost.
- 5) A performance analysis and comparisons with SVD method are conducted based on partially theoretical development followed by simulated experiments.

The rest of the paper is organized as follows. The MNS concept and its implementation are briefly reviewed in Section II. The GMNS concept and the GMNS-based batch algorithms for MSA and PSA/PCA are proposed in Sections III and IV, respectively. Adaptive versions of the proposed algorithms are then developed in Section V, with some details on the numerical complexity. An asymptotic performance analysis is given in Section VI. The performance of the proposed batch and adaptive algorithms in batch and adaptive systems in Section VII.

## II. MINIMUM NOISE SUBSPACE: A REVIEW

Let us consider a general linear system with  $p$  inputs and  $n$  outputs ( $p < n$ ), which obeys the following model for the input–output relationship:

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

where  $\mathbf{x}(t) \in \mathbb{C}^n$  is the observation vector,  $\mathbf{A} \in \mathbb{C}^{n \times p}$  is the system matrix having full column rank,  $\mathbf{s}(t) \in \mathbb{C}^p$  is random source vector, and  $\mathbf{n}(t) \in \mathbb{C}^n$  is the additive white noise vector with unknown variance  $\sigma^2$ . The data covariance matrix is then given by

$$\mathbf{R}_{xx} = E\{\mathbf{x}(t)\mathbf{x}(t)^H\} = \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H + \sigma^2\mathbf{I}, \quad (2)$$

where  $\mathbf{R}_{ss}$  is the source covariance matrix having full rank.

It is of interest to compute the minor (or principal) subspace of  $\mathbf{R}_{ss}$  for that the MNS method was proposed to achieve a fast computation in a parallel manner [12]. In particular, the concept of *properly connected sequence* (PCS) is defined and the system is re-organized into  $n - p$  subsystems based on a selected PCS. Then, the minor subspace will be efficiently estimated by computing the least (noise) eigenvector of each subsystem. The PCS concept is to guarantee that the noise vectors computed from the subsystems form a basis of the noise subspace. In the following, we will review the PCS concept and MNS computation.

### A. Properly Connected Sequence

Denote the  $n$  system outputs by a set of members  $m_1, m_2, \dots, m_n$ . Consider a sequence of  $n - p$  subsets of outputs, wherein each subset  $i$  contains  $p + 1$  members and is denoted by the  $(p + 1)$ -tuple  $t_i = (m_{i_1}, m_{i_2}, \dots, m_{i_{p+1}})$ ,  $i = 1, \dots, n - p$ . This sequence is said to be *properly connected*, if the following conditions are satisfied:

$$\begin{aligned} \{m_{i_1}, \dots, m_{i_p}\} &\subset \{m_{j_k} \mid j < i, 1 \leq k \leq p + 1\}, \\ m_{i_{p+1}} &\notin \{m_{j_k} \mid j < i, 1 \leq k \leq p + 1\}. \end{aligned}$$

These conditions mean that each tuple in the sequence has  $p$  members in common with its preceding tuples, plus with another member not so. For instance, if we consider a system with  $p = 2$  inputs and  $n = 6$  outputs, the following sequence of  $n - p = 4$  tuples is a PCS:

$$\begin{aligned} t_1 &= (m_1, m_2, m_3); t_2 = (m_1, m_2, m_4); \\ t_3 &= (m_2, m_3, m_5); t_4 = (m_2, m_5, m_6). \end{aligned}$$

### B. MNS Implementation

Now, given a PCS of  $n - p$  tuples of system outputs, the (partial) observation vector  $\mathbf{x}_i(t) = [x_{i_1}(t), x_{i_2}(t), \dots, x_{i_{p+1}}(t)]^T$  of  $p + 1$  outputs corresponding to the  $i$ -th tuple has the following covariance matrix:

$$\mathbf{R}_i = E\{\mathbf{x}_i(t)\mathbf{x}_i^H(t)\} = \mathbf{A}_i\mathbf{R}_{ss}\mathbf{A}_i^H + \sigma^2\mathbf{I}, \quad (3)$$

where  $\mathbf{A}_i$  is the response matrix of the corresponding subsystem. Each subsystem in (3) has a noise subspace of minimum dimension (equal to one), suggesting the name minimum noise subspace (MNS).

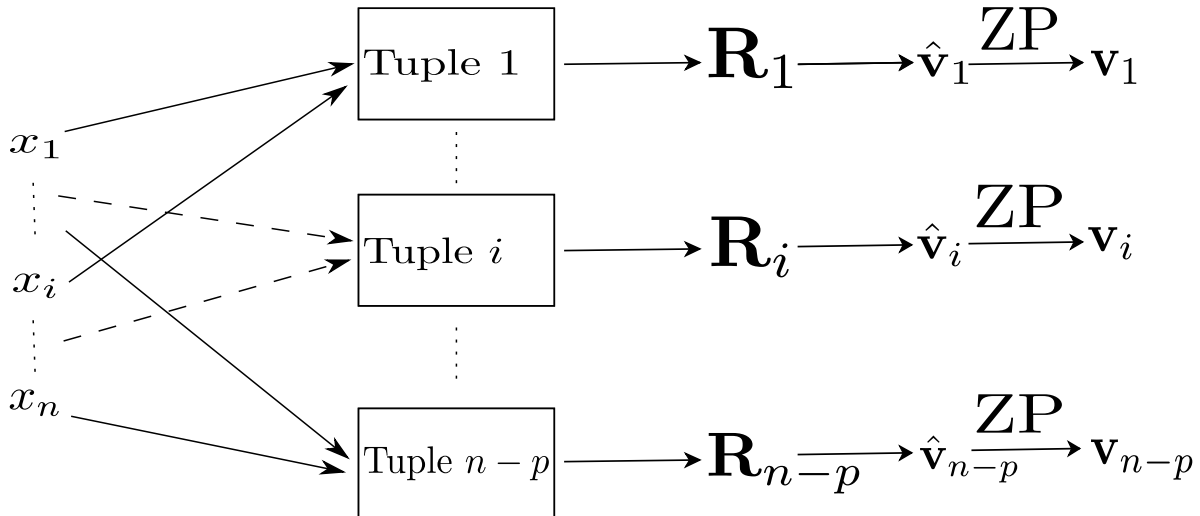


Fig. 1. Illustration of MNS implementation.

From each  $\mathbf{R}_i^1$ , a noise vector  $\mathbf{v}_i$  is constructed by first computing the least eigenvector  $\hat{\mathbf{v}}_i$  of  $\mathbf{R}_i$ , and then zero-padding (ZP) it according to

$$\mathbf{v}_i(j) = \begin{cases} 0, & \text{if the } j\text{-th output of the system} \\ & \text{does not belong to the } i\text{-th tuple,} \\ \hat{\mathbf{v}}_i(j'), & \text{if the } j\text{-th output of the system} \\ & \text{is the } j'\text{-th entry of } i\text{-th tuple,} \end{cases} \quad (4)$$

for  $1 \leq j \leq n$ .

It is proved in [12], [13] that the resulting set of noise vectors  $\{\mathbf{v}_i\}$  for  $1 \leq i \leq n-p$  form a basis of the noise subspace. Figure 1 illustrates an MNS implementation.

### III. MINOR SUBSPACE ANALYSIS USING GMNS

As stated in Section I, the objective of this paper is to exploit the available computing architecture, when having access to exactly  $K$  parallel units, to reduce the computational cost for extracting the minor or principal subspaces of  $\mathbf{R}_{xx}$ . We approach this problem by generalizing the MNS method.

It is observed that, in the MNS method, each noise vector is estimated by the use of a minimum number of system outputs (i.e.,  $p+1$ ) which might lead to a non-negligible performance loss if

<sup>1</sup>In practice,  $\mathbf{R}_i$  is replaced by its sample averaged estimate  $\hat{\mathbf{R}}_i = \frac{1}{T} \sum_t \mathbf{x}_i(t) \mathbf{x}_i^H(t)$ , with  $T$  being the sample size.

$n \gg p$ . In addition, to achieve the parallel computation of the noise vectors,  $n - p$  computing units are needed, a number which depends on the impinging source number  $p$  and is usually a non-controllable system parameter.

In this paper, we propose the GMNS method, which is a generalization of the MNS method, that overcomes the above-mentioned shortcomings. Given  $K$  computing units, we need to compute the  $n - p$  noise vectors.

Write  $n - p = dK + r$ , where  $d$  and  $r$  are integers and  $0 \leq r < K$ . For simplicity, we assume that  $r = 0$  (i.e.,  $(n - p)/K$  is integer-valued). We now propose the concept of generalized properly connected sequence (GPCS), generalizing the PCS concept used in the MNS method.

**Definition 1.** *The GPCS is a sequence of  $K$   $(p + d)$ -tuples  $t_i = (m_{i_1}, \dots, m_{i_{p+d}})$  for  $1 \leq i \leq K$  which satisfy the following conditions:*

$$\{m_{i_1}, \dots, m_{i_p}\} \subset \{m_{j_k} \mid j < i, 1 \leq k \leq p + d\}, \quad (5)$$

$$\{m_{i_{p+1}}, \dots, m_{i_{p+d}}\} \not\subset \{m_{j_k} \mid j < i, 1 \leq k \leq p + d\}. \quad (6)$$

In other words, each tuple in the sequence has  $p$  members in common with its preceding tuples plus  $d$  other members not so.

Given a GPCS of  $K$   $(p + d)$ -tuples of the outputs, the noise vectors are computed as follows. First, for each subsystem, we compute the covariance matrix  $\mathbf{R}_i$ , with size of  $(p + d) \times (p + d)$ , and hence its  $d$  least eigenvectors, represented by matrix  $\hat{\mathbf{V}}_i$ . Then, we construct the desired noise submatrix  $\mathbf{V}_i$  according to<sup>2</sup>

$$\mathbf{V}_i(j, :) = \begin{cases} 0, & \text{if the } j\text{-th output of the system} \\ & \text{does not belong to the } i\text{-th tuple,} \\ \hat{\mathbf{V}}_i(j', :), & \text{if the } j\text{-th output of the system} \\ & \text{is the } j'\text{-th entry of } i\text{-th tuple,} \end{cases} \quad (7)$$

Finally, we concatenate the submatrices  $\mathbf{V}_i$  for  $1 \leq i \leq K$  to have the global noise matrix  $\mathbf{V} = [\mathbf{V}_1, \dots, \mathbf{V}_K]$  whose columns form a basis of the noise subspace under the conditions given in the following theorem.

<sup>2</sup>In the sequel,  $\mathbf{V}(j, :)$  refers to  $j$ -th row vector of  $\mathbf{V}$ .

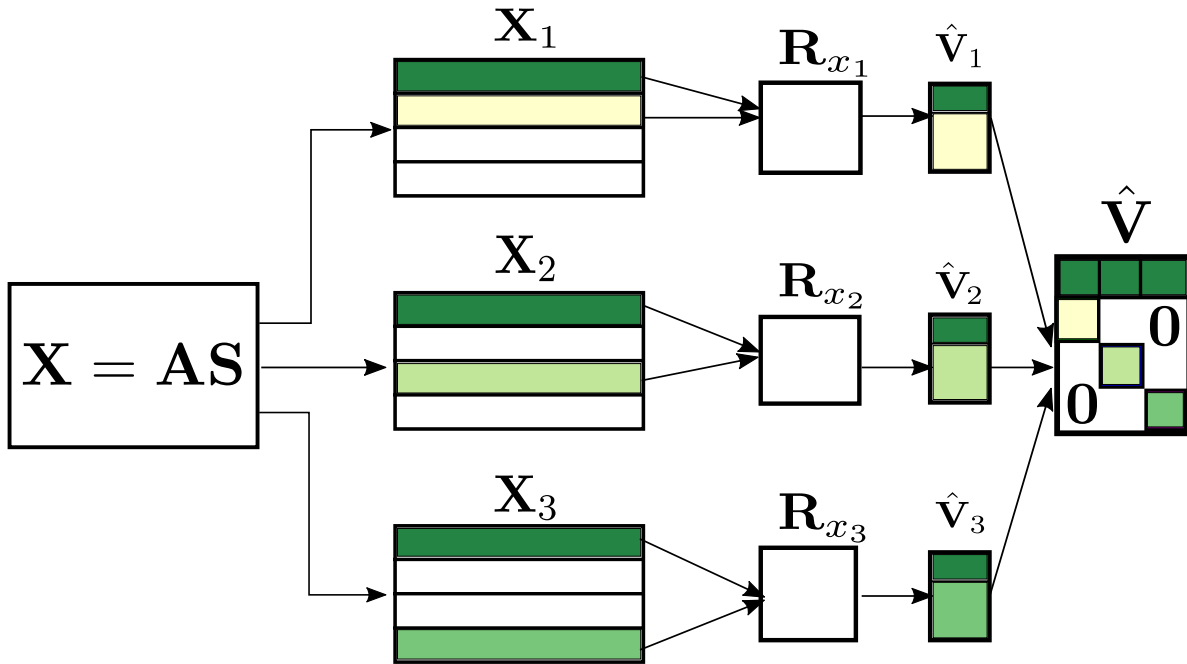


Fig. 2. GMNS for MSA, where  $K = 3$  subsystems are considered. In this example, the green (bold) part represents the  $p$  outputs shared by three subsystems.

**Theorem 1.** *Under the assumption that every  $p$  rows of  $\mathbf{A}$  are linearly independent, the noise matrix  $\mathbf{V}$  has full column rank (i.e.,  $\text{rank}(\mathbf{V}) = n - p$ ) and hence its columns span the desired noise subspace of the data covariance matrix  $\mathbf{R}_{xx}$ .*

The proof of Theorem 1 is given in Appendix A. Figure 2 provides an illustration of the GMNS concept and Table I summarizes steps of the method.

The main advantage of our proposed GMNS method is the reduction of the computational cost, as compared to the standard method. In particular, GMNS requires  $O((p + (n - p)/K)^2 T)$  flops for the computation of the subsystem covariance matrices plus  $O((p + (n - p)/K)^2 (n - p)/K)$  flops for the estimation of the least eigenvectors. Whereas, the standard method requires  $O(n^2 T)$  flops for the estimation of the global covariance matrix costs plus  $O(n^2 (n - p))$  flops for the extraction of the noise vectors. Obviously, if  $n \gg p$ , the overall cost is almost reduced by a factor  $K^2$  for the covariance matrix estimation and a factor of  $K^3$  for the noise subspace estimation.

**Remarks:**

- 1) In the general case,  $(n - p)/K$  is non integer-valued; that is,  $0 < r < K$  for  $r$  in  $n - p = dK + r$ . In that case, we will use  $r$  tuples of length  $p + d'$ , with  $d' = d + 1$ , and



$K - r$  tuples of length  $p + d$ .

- 2) Note that GPCS is just a practical way to guarantee that the obtained vectors form a basis of the desired subspace. In other words, the GPCS does not represent necessary conditions to meet but only sufficient conditions.
- 3) For large dimensional systems where  $n \gg p$ , using only  $p + 1$  system outputs as in the original MNS to compute a noise vector may result in non-negligible performance loss. Now if  $\lfloor (n - p)/K \rfloor = d$  is relatively large, we will instead use  $p + d + 1$  outputs to estimate a given noise vector which improves its estimation accuracy. (This accuracy is illustrated in Figures 5 and 6).
- 4) In [18], a fast subspace estimation method was proposed, exploiting the spatial whiteness of the additive noise. In this paper, we exploit this property together with parallel computing to achieve much higher numerical gain.

#### IV. PRINCIPAL SUBSPACE/COMPONENT ANALYSIS USING GMNS

The original MNS was dedicated to the MSA problem. Here, we introduce a new GMNS method to deal with the PSA problem using  $K$  subsystems in a parallel scheme. In particular, we proposed two algorithms for overlapping and non-overlapping subsystems respectively.

##### A. Principal Subspace Analysis

###### 1) Subsystems without overlapping parts:

Let assume that we have a large dimensional system such that  $l = n/K > p$  and, for simplicity, integer-valued. We divide the  $n$  system outputs into  $K$  subsystems of length  $n/K$  each represented by

$$(m_{(i-1)l+1}, \dots, m_{il}), \quad i = 1, \dots, K.$$

Now, for each subsystem, we compute the corresponding covariance matrix  $\mathbf{R}_i$  and its principal subspace matrix  $\mathbf{W}_i = \mathbf{A}_i \mathbf{Q}_i$ , where  $\mathbf{Q}_i$  is an unknown nonsingular matrix of size  $p \times p$ .

To have a global estimate of the signal subspace (i.e., a matrix  $\mathbf{W} = \mathbf{A} \mathbf{Q}$  of size  $n \times p$  where  $\mathbf{Q}$  is any  $p \times p$  nonsingular matrix), we need to get rid of the unknown matrices  $\mathbf{Q}_i$ . For that, we exploit the fact that all subsystems receive the same source vector  $\mathbf{S}$  of size  $p \times T$ , that is,

$$\mathbf{X}_i = \mathbf{A}_i \mathbf{S} + \mathbf{N}_i, \quad i = 1, \dots, K, \quad (8)$$

where  $\mathbf{N}_i$  is noise affecting the  $i$ -th subsystem. Define

$$\mathbf{S}_i = \mathbf{W}_i^\# \mathbf{X}_i, \quad (9)$$

where  $\#$  denotes the pseudo-inverse operator<sup>3</sup>. Then, thanks to (8), we have

$$\mathbf{S}_i = \mathbf{Q}_i^{-1} \mathbf{S} + \mathbf{W}_i^\# \mathbf{N}_i, \quad i = 1, \dots, K, \quad (10)$$

and hence, in the noiseless case, it can be shown that

$$\mathbf{S}_i = \mathbf{T}_i \mathbf{S}_1,$$

where  $\mathbf{T}_i = \mathbf{Q}_i^{-1} \mathbf{Q}_1$ . The matrix  $\mathbf{T}_i$  can be estimated by solving the least square problem

$$\min_{\mathbf{T}_i} \|\mathbf{S}_i - \mathbf{T}_i \mathbf{S}_1\|_2^2$$

to obtain the solution

$$\hat{\mathbf{T}}_i = \mathbf{S}_i \mathbf{S}_1^\#. \quad (11)$$

Finally, the principal subspace weight matrix is obtained as

$$\mathbf{W} = [\mathbf{W}_1^T, (\mathbf{W}_2 \mathbf{T}_2)^T, \dots, (\mathbf{W}_K \mathbf{T}_K)^T]^T \simeq \mathbf{A} \mathbf{Q}_1 \quad (12)$$

In the noisy case, the estimate of the principal subspace weight matrix in (12) is *biased* due to the effect of the noise term on the estimation of the matrix  $\mathbf{T}_i$  in (11). In fact, (11) can be rewritten as

$$\hat{\mathbf{T}}_i = \left( \frac{\mathbf{S}_i \mathbf{S}_1^H}{T} \right) \left( \frac{\mathbf{S}_1 \mathbf{S}_1^H}{T} \right)^{-1} \quad (13)$$

$$\simeq (\mathbf{Q}_i^{-1} \hat{\mathbf{R}}_{ss} \mathbf{Q}_1^{-H}) (\mathbf{Q}_1^{-1} \hat{\mathbf{R}}_{ss} \mathbf{Q}_1^{-H} + \sigma^2 \mathbf{I})^{-1}. \quad (14)$$

Here we used the fact that the subsystems being non-overlapping, their noise terms are uncorrelated (spatially white noise assumption) and the fact that  $\mathbf{W}_i$  are unitary matrices leading to  $E[\mathbf{W}_1^H \mathbf{n}_1(t) \mathbf{n}_1^H(t) \mathbf{W}_1] = \sigma^2 \mathbf{I}$ . Because of the additive term  $\sigma^2 \mathbf{I}$ ,  $\hat{\mathbf{T}}_i$  deviates from its desired value and leads to an estimation bias for the global weight matrix, especially, at low SNRs.

<sup>3</sup>Most subspace estimation methods compute an orthonormal basis of the desired subspace (see, e.g., [19]) in which case we have  $\mathbf{W}_i^\# = \mathbf{W}_i^H$ .

To overcome this problem, we replace the previous estimate of  $\mathbf{T}_i$  by the following asymptotically unbiased estimate

$$\begin{aligned}\tilde{\mathbf{T}}_i &= \left( \frac{\mathbf{S}_i \mathbf{S}_1^H}{T} \right) \left( \frac{\mathbf{S}_1 \mathbf{S}_1^H}{T} - \hat{\sigma}^2 \mathbf{I} \right)^{-1} \\ &= (\mathbf{W}_i^H \mathbf{R}_{i,1} \mathbf{W}_1) (\mathbf{W}_1^H \mathbf{R}_1 \mathbf{W}_1 - \hat{\sigma}^2 \mathbf{I})^{-1},\end{aligned}\quad (15)$$

$$\hat{\sigma}^2 = [\text{Tr}(\mathbf{R}_1) - \text{Tr}(\mathbf{W}_1^H \mathbf{R}_1 \mathbf{W}_1)] / (l - p), \quad (16)$$

where  $\mathbf{R}_{i,1} = E[\mathbf{x}_i(t) \mathbf{x}_1^H(t)]$ .

The noise power estimate in (16) comes from the fact that

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{A}_1 \mathbf{R}_{ss} \mathbf{A}_1^H + \sigma^2 \mathbf{I} \\ &= \mathbf{U}_s (\mathbf{\Lambda}_s + \sigma^2 \mathbf{I}) \mathbf{U}_s^H + \sigma^2 \mathbf{U}_n \mathbf{U}_n^H\end{aligned}\quad (17)$$

$$\mathbf{W}_1^H \mathbf{R}_1 \mathbf{W}_1 = \mathbf{W}_1^H \mathbf{U}_s (\mathbf{\Lambda}_s + \sigma^2 \mathbf{I}) \mathbf{U}_s^H \mathbf{W}_1 \quad (18)$$

The first equality refers to the eigen-decomposition of  $\mathbf{R}_1$ ; the columns of  $\mathbf{U}_s$  span the signal subspace of  $\mathbf{R}_1$  while those of  $\mathbf{U}_n$  span the noise subspace of  $\mathbf{R}_1$ . The second equality holds because  $\mathbf{W}_1^H \mathbf{U}_n = \mathbf{0}$ . Now, since  $\mathbf{W}_1^H \mathbf{U}_s$  is unitary, we have  $\text{Tr}(\mathbf{R}_1 - \mathbf{W}_1^H \mathbf{R}_1 \mathbf{W}_1) = (l - p) \sigma^2$ , and hence the result (16).

The above biased and unbiased algorithms are referred to as GMNS-N-PSA (N stands for non-overlapping) and GMNS-NU-PSA (NU stands for non-overlapping and unbiased), respectively. They are summarized in Table II. Figure 3 exemplifies the steps described in this section.

## 2) Subsystems with overlapping parts:

In the non-overlapping case, we assumed that  $n/K > p$ . To relax this assumption and extend our method to cover also the case<sup>4</sup> where  $n/K < p$ , we consider here overlapping subsystems of size  $p + q$  sharing  $q$  system outputs, and represented by the  $K$  tuples. For example, here, we choose all subsystems which overlap with the first one:

$$\begin{aligned}&(m_1, \dots, m_q, m_{q+1}, \dots, m_{p+q}), \\ &(m_1, \dots, m_q, m_{(i-1)d+1}, \dots, m_{id}), \quad i = 2, \dots, K.\end{aligned}$$

<sup>4</sup>When using overlapping subsystems, we can deal with both situations, i.e.,  $n/K \geq p$  and  $n/K < p$ , while in the previous non-overlapping case, it is necessary to have  $n/K > p$ .

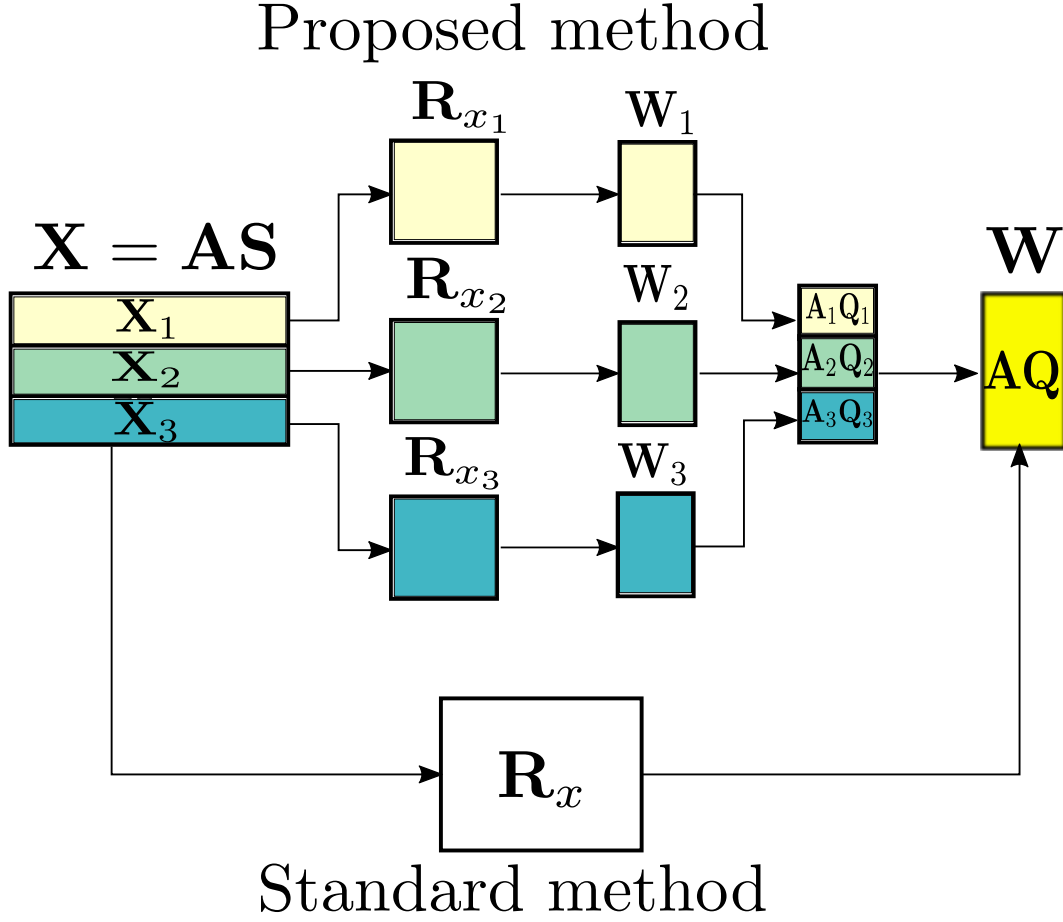


Fig. 3. GMNS for PSA with non-overlapping parts;  $K = 3$ .

In other words, the  $q$  first members of the first subsystem are the  $q$  first members of the  $i$ -th subsystem, for  $i = 2, \dots, K$ . For simplicity, we assume that  $d = (n - p)/K$  is integer-valued. Now, for each subsystem, we compute the covariance matrix  $\mathbf{R}_i$  and its corresponding weight matrix  $\mathbf{W}_i$  which can be written as

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{W}_i^{\text{olap}} \\ \mathbf{W}'_i \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i^{\text{olap}} \\ \mathbf{A}'_i \end{bmatrix} \mathbf{Q}_i. \quad (19)$$

To get rid of the matrices  $\mathbf{Q}_i$ , one exploits the overlap between the first subsystem and the  $i$ -th subsystem by assuming that any  $p \times p$  submatrix of  $\mathbf{A}$  has full rank. In that case, the global weight matrix is estimated as

$$\mathbf{W} = \left[ \tilde{\mathbf{W}}_1^T, \tilde{\mathbf{W}}_2^T, \dots, \tilde{\mathbf{W}}_K^T \right]^T, \quad (20)$$

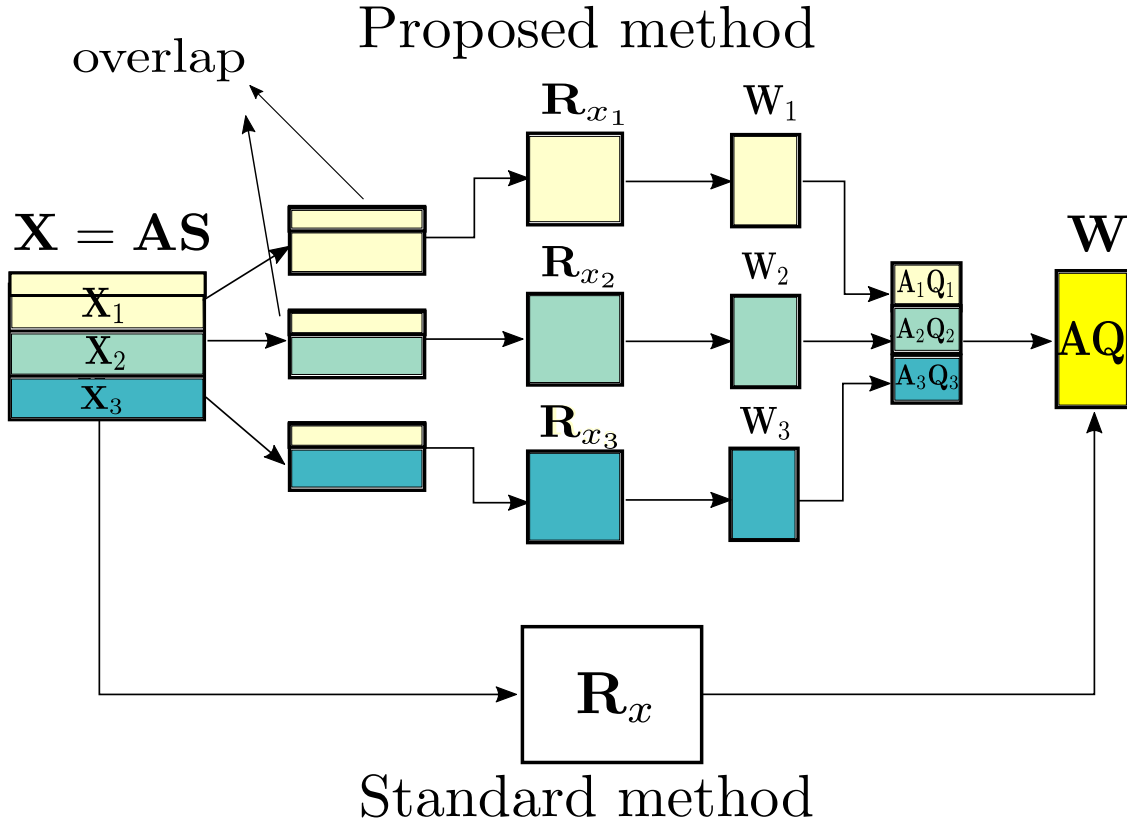


Fig. 4. GMNS for PSA with overlapping parts;  $K = 3$ .

where

$$\tilde{W}_1 = W_1, \quad (21)$$

$$\tilde{W}_i = W_i T_i, \quad i = 2, \dots, K, \quad (22)$$

with

$$T_i = (W_i^{\text{olap}})^{\#} (W_1^{\text{olap}}). \quad (23)$$

This algorithm is referred to as GMNS-O-PSA (where O stands for overlapping) and is summarized in Table III. Figure 4 illustrates our description.

Similarly to the GMNS-MSA, the main advantage of the proposed GMNS-based PSA algorithms resides in its reduced computational cost. Indeed, GMNS-NU-PSA costs  $O((n/K)^2 p + p^2(n/K + p))$  flops for the computation of the  $p$  signal subspace vectors and  $O(2(n/K)^2 T)$  flops for the computation of the covariance matrices  $R_i$  and the correlation matrices  $R_{i,1}$ , for  $i = 1, \dots, K$ . This overall cost is approximately  $K^2$  less than the cost of a direct computation of

TABLE I  
SUMMARY OF GMNS FOR MSA

<p>Input: the captured data <math>\mathbf{X}</math></p> <ol style="list-style-type: none"> <li>1) Extract subsystems <math>\mathbf{X}_i</math> as described in III</li> <li>2) Compute the covariance matrix of each subsystem <math>\mathbf{R}_i</math>: <math>\mathbf{R}_i = \frac{1}{T} \mathbf{X}_i \mathbf{X}_i^H</math>.</li> <li>3) Extract <math>d</math> least eigenvectors of <math>\mathbf{R}_i</math>: <math>\hat{\mathbf{V}}_i = \text{eig}(\mathbf{R}_i)</math>.</li> <li>4) Construct the desired noise matrix <math>\mathbf{V}_i</math>, where</li> </ol> $\mathbf{V}_i(j, :) = \begin{cases} 0, & \text{if the } j\text{-th output of the system does not belong to the } i\text{-th tuple,} \\ \hat{\mathbf{V}}_i(j', :), & \text{if the } j\text{-th output of the system is the } j'\text{-th entry of } i\text{-th tuple.} \end{cases}$ <p>Output: the minor subspace weight matrix: <math>\mathbf{V} = [\mathbf{V}_1, \dots, \mathbf{V}_K]</math>.</p>
--

TABLE II  
SUMMARY OF GMNS FOR PSA: THE CASE OF NON-OVERLAPPING SUBSYSTEMS

<p>Input: the captured data <math>\mathbf{X} = [\mathbf{X}_1^T, \dots, \mathbf{X}_K^T]^T</math></p> <ol style="list-style-type: none"> <li>1) Compute the covariance matrix <math>\mathbf{R}_i</math> of each subsystem: <math>\mathbf{R}_i = \frac{1}{T} \mathbf{X}_i \mathbf{X}_i^H</math>.</li> <li>2) Extract <math>p</math> principal eigenvectors of <math>\mathbf{R}_i</math>: <math>\mathbf{W}_i = \text{eig}(\mathbf{R}_i, p)</math>.</li> <li>3) Compute <math>\tilde{\mathbf{T}}_i</math> <ul style="list-style-type: none"> <li>• Biased case: <math>\mathbf{S}_i = \mathbf{W}_i^\# \mathbf{X}_i</math>, <math>\tilde{\mathbf{T}}_i = \mathbf{S}_i \mathbf{S}_i^\#</math>.</li> <li>• Unbiased case: <math>\hat{\sigma}^2 = [\text{Tr}(\mathbf{R}_i) - \text{Tr}(\mathbf{W}_i^H \mathbf{R}_i \mathbf{W}_i)] / (l - p)</math>, <math>\tilde{\mathbf{T}}_i = \left( \frac{\mathbf{S}_i \mathbf{S}_i^H}{T} \right) \left( \frac{\mathbf{S}_i \mathbf{S}_i^H}{T} - \hat{\sigma}^2 \mathbf{I} \right)^{-1}</math>.</li> </ul> </li> </ol> <p>Output: the principal subspace <math>\mathbf{W} = [\mathbf{W}_1^T, (\mathbf{W}_2 \tilde{\mathbf{T}}_2)^T, \dots, (\mathbf{W}_K \tilde{\mathbf{T}}_K)^T]^T</math>.</p>
---

the signal subspace using the global covariance matrix, which takes  $O(n^2(T + p))$  flops. On the other hand, algorithm GMNS-O-PSA costs  $O((p + q)^2 p + p^2(2p + q))$  flops for the computation of the  $p$  signal subspace vectors and  $O((p + q)^2 T)$  flops for the parallel computation of the covariance matrices. If  $n \gg p$ ,  $n \gg K$  and  $T \gg 1$ , we have  $q \approx n/K$ , in which case GMNS-O-PSA is slightly cheaper than GMNS-NU-PSA since it does not require the computation of the correlation matrices<sup>5</sup>  $\mathbf{R}_{i,1}$ .

<sup>5</sup>For certain applications, e.g., radioastronomy, the global covariance matrix is available or computed for other needs, in which case GMNS-NU-PSA becomes more advantageous than GMNS-O-PSA in term of numerical cost.

TABLE III  
SUMMARY OF GMNS FOR PSA: THE CASE OF OVERLAPPING SUBSYSTEMS

<p>Input: the captured data <math>\mathbf{X}</math></p> <ol style="list-style-type: none"> <li>1) Extract subsystems <math>\mathbf{X}_i</math> as described in IV-A2</li> <li>2) Compute the covariance matrix <math>\mathbf{R}_i</math> of each subsystem: <math>\mathbf{R}_i = \frac{1}{T} \mathbf{X}_i \mathbf{X}_i^H</math>.</li> <li>3) Extract <math>p</math> principal eigenvectors of <math>\mathbf{R}_i</math>: <math>\mathbf{W}_i = \text{eig}(\mathbf{R}_i, p)</math>.</li> <li>4) Extract the overlap between the first and the <math>i</math>-th subsystems: <math>\mathbf{W}_1^{\text{olap}}(t), \mathbf{W}_i^{\text{olap}}(t)</math>.</li> <li>5) Compute matrices <math>\hat{\mathbf{T}}_i, i = 2, \dots, K</math>: <math>\mathbf{T}_i = (\mathbf{W}_i^{\text{olap}})^\dagger (\mathbf{W}_1^{\text{olap}})</math> and the principal weight matrix as <math>\tilde{\mathbf{W}}_1 = \mathbf{W}_1</math>, <math>\tilde{\mathbf{W}}_i = \mathbf{W}_i \mathbf{T}_i</math>.</li> </ol> <p>Output: the principal subspace <math>\mathbf{W} = [\tilde{\mathbf{W}}_1^T, \tilde{\mathbf{W}}_2^T, \dots, \tilde{\mathbf{W}}_K^T]^T</math>.</p>
---

TABLE IV  
SUMMARY OF GMNS FOR PCA

<p>Input: the principal subspace <math>\mathbf{W}</math></p> <ol style="list-style-type: none"> <li>1) Compute the expression <math>\mathbf{W}^\# \mathbf{R}_{xx} \mathbf{W}</math>.</li> <li>2) Solve diagonalization problem to find <math>\tilde{\mathbf{Q}}^{-1}</math> and <math>\{\lambda_i\}</math></li> <li>3) Extract <math>p</math> the principal eigenvectors as: <math>\mathbf{U}_s = \mathbf{W} \tilde{\mathbf{Q}}^{-1}</math>.</li> </ol> <p>Output: the principal eigenpairs <math>\mathbf{U}_s</math> and <math>\{\lambda_i\}_{i=1:p}</math></p>
--

### B. Principal Component Analysis using GMNS

After estimating the principal subspace as presented in the above sections, we can further extract principal eigenvalues and eigenvectors of the covariance matrix  $\mathbf{R}_{xx}$  as follows. First, we note that

$$\mathbf{W} = \mathbf{A} \mathbf{Q} = \mathbf{U}_s \tilde{\mathbf{Q}}, \quad (24)$$

where columns of  $\mathbf{U}_s$  are the  $p$  principal eigenvectors of  $\mathbf{R}_{xx}$  and  $\tilde{\mathbf{Q}}$  is a  $p \times p$  non-singular matrix. Now, consider the following expression

$$\mathbf{W}^\# \mathbf{R}_{xx} \mathbf{W} = \tilde{\mathbf{Q}}^{-1} (\mathbf{U}_s^H \mathbf{R}_{xx} \mathbf{U}_s) \tilde{\mathbf{Q}} \quad (25)$$

$$= \tilde{\mathbf{Q}}^{-1} \begin{bmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_p \end{bmatrix} \tilde{\mathbf{Q}} \quad (26)$$

where  $\lambda_1, \dots, \lambda_p$  are the principal eigenvalues of  $\mathbf{R}_{xx}$ . Therefore, the principal eigenvectors and eigenvalues can be found by computing matrix  $\tilde{\mathbf{Q}}$  which diagonalizes the matrix  $\mathbf{W}^\# \mathbf{R}_{xx} \mathbf{W}$

## V. ADAPTIVE GMNS-BASED ALGORITHMS

In this section, we are interested in estimating the minor and principal subspaces at each time index  $t$  from streaming observations  $\{\mathbf{x}(t)\}_{t \geq 1}$ .

### A. Minor Subspace Tracking using GMNS

Thanks to the parallel structure of GMNS-MSA, the conversion from batch system to adaptive one is quite simple. In practice, to track the underlying minor subspace, we replace the step 2 in Table I (computing minor subspace from correlation matrix using SVD) by existing minor subspace tracking (MST) algorithms while keeping the remaining steps unchanged. For example, we integrate two algorithms FOOja [20] and FDPM [11] into our scheme. Note that, the covariance matrix estimation step (i.e., step 1 in Table I) is not required in these algorithms. As a result, our framework provides a way to reduce the complexity of the algorithms by a factor of  $K$ . This algorithm is referred to as GMNS-MST and is summarized in Table V.

Because of the way we construct the desired noise matrix (i.e., step 3 in Table V), the cost of the whole procedure equals the computational complexity of the tracking algorithms reduced by a factor of  $K$ . For example, it costs  $O(np/K)$  if FOOja or FDPM is used.

### B. Principal Subspace Tracking using GMNS

#### 1) Subsystems without overlapping parts:

We observe that matrix  $\mathbf{T}_i$  can be expressed by

$$\mathbf{T}_i = \mathbf{W}_i^\# (\mathbf{X}_i \mathbf{X}_1^\#) \mathbf{W}_1. \quad (27)$$



By this way, we can track the principal subspace  $\mathbf{W}_i$  of each subsystem and then compute the global weight matrix  $\mathbf{W}$  as in (12). However, calculating the term  $\mathbf{X}_i\mathbf{X}_1^\#$  in full scale is expensive and not suitable for adaptive scheme. Therefore, we propose to use a sliding window technique to tackle this problem. Denote by  $N$  the window size. At time instant  $t$ , the subsystem  $\mathbf{X}_i(t)$  can be written as

$$\mathbf{X}_i(t) = [\mathbf{x}_i(t - (N - 1)) \dots \mathbf{x}_i(t)]. \quad (28)$$

For principal subspace tracking, we will use the sliding window version of the OPAST algorithm [21]. Now to update  $\mathbf{T}_i(t)$  efficiently, we can rewrite the term  $\mathbf{X}_i(t)\mathbf{X}_1^\#(t)$  as

$$\mathbf{X}_i(t)\mathbf{X}_1^\#(t) = \mathbf{P}_i(t)\mathbf{M}^{-1}(t) \quad (29)$$

where

$$\mathbf{P}_i(t) = \mathbf{X}_i(t)\mathbf{X}_1^H(t), \quad (30)$$

$$\mathbf{M}(t) = \mathbf{X}_1(t)\mathbf{X}_1^H(t). \quad (31)$$

Then we get

$$\begin{aligned} \mathbf{P}_i(t) &= \sum_{\tau=0}^{N-1} \mathbf{x}_i(t-\tau)\mathbf{x}_1^H(t-\tau) \\ &= \mathbf{P}_i(t-1) + \mathbf{x}_i(t)\mathbf{x}_1^H(t) - \mathbf{x}_i(t-N)\mathbf{x}_1^H(t-N) \end{aligned} \quad (32)$$

$$\begin{aligned} \mathbf{M}(t) &= \sum_{\tau=0}^{N-1} \mathbf{x}_1(t-\tau)\mathbf{x}_1^H(t-\tau) \\ &= \mathbf{M}(t-1) + \mathbf{x}_1(t)\mathbf{x}_1^H(t) - \mathbf{x}_1(t-N)\mathbf{x}_1^H(t-N) \end{aligned} \quad (33)$$

Since  $\mathbf{M}(t)$  has a rank-2 update structure, it can be efficiently inverted by applying the matrix inversion lemma to give the expression of  $\mathbf{M}^{-1}(t)$ . However, substituting (32) and (33) into (29) and hence into (27) still includes expensive matrix-matrix multiplications. Note that many fast tracking algorithms have rank-1 update form (e.g., OPAST [10]) as follows:

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{p}(t)\mathbf{q}^H(t). \quad (34)$$

Thus, we can use this observation to compute (27) recursively with only matrix-vector multiplications. To initialize  $\mathbf{P}_i(0)$  and  $\mathbf{M}(0)$ , we can either choose them arbitrarily or use the  $N$  first snapshots in a batch way.

This algorithm is referred to as GMNS-N-PST and summarized in Table VI.

TABLE V  
SUMMARY OF GMNS FOR MINOR SUBSPACE TRACKING

<p>Initialize: Initialization of the tracking algorithm</p> <p>Input: the captured data at snapshot <math>t</math>-th <math>\mathbf{x}(t)</math></p> <ol style="list-style-type: none"> <li>1) Extract subsystems <math>\mathbf{x}_i(t)</math> as described in III</li> <li>2) Tracking the minor subspace of each subsystem using some existing algorithms (e.g. FOOja, FDPm):</li> </ol> $\hat{\mathbf{V}}_i(t) = \text{TrackingAlgorithm}(\mathbf{V}_i(t-1), \mathbf{x}_i(t)).$ <ol style="list-style-type: none"> <li>3) Construct the desired noise matrix <math>\mathbf{V}_i(t)</math>: <math>\mathbf{V}_i(j, :) = \begin{cases} 0, &amp; \text{if the } j\text{-th output of the system does not belong to the } i\text{-th tuple,} \\ \hat{\mathbf{V}}_i(j', :), &amp; \text{if the } j\text{-th output of the system is the } j'\text{-th entry of } i\text{-th tuple.} \end{cases}</math></li> </ol> <p>Output: the minor subspace <math>\mathbf{V}(t) = [\mathbf{V}_1(t), \dots, \mathbf{V}_K(t)]</math></p>
---

**Remark:** We have observed that in the adaptive scheme, the improvement of tracking the principal subspace in the unbiased case is negligible as compared to the biased case. Thus, we only present the latter due to its simplicity.

### 2) Subsystems with overlapping parts:

Now, we derive the subspace tracking method for subsystems with overlapping parts. Similar to the GMNS-N-MST, at each time instant, the adaptive version of GMNS-O-PSA simply substitutes computing the principal subspace from the correlation matrix using SVD (i.e. step 2 and 3) by the existing principal subspace tracking algorithms and then keep the remaining steps. Again, a similar observation for computing  $\mathbf{T}_i$  efficiently in the non-overlapping case can also apply here. This algorithm is referred to as GMNS-O-PST and summarized in Table VII.

### C. Principal Eigenvector Tracking using GMNS

In this section, we present an algorithm to extract the principal eigenvectors from the estimated principal subspace. This is an adaptive version of the GMNS for PCA, as described in Section IV-B.

According to (25), at time instant  $t$ , the following diagonalization is performed:

$$\mathbf{W}^\#(t)\mathbf{R}_{xx}(t)\mathbf{W}(t) = \tilde{\mathbf{Q}}^{-1}(t) \begin{bmatrix} \lambda_1(t) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_p(t) \end{bmatrix} \tilde{\mathbf{Q}}(t). \quad (35)$$

TABLE VI  
SUMMARY OF GMNS FOR PRINCIPAL SUBSPACE TRACKING: THE CASE OF NON-OVERLAPPING SUBSYSTEMS

<p>Initialize: initialization of the tracking algorithm, <math>\mathbf{M}(0)</math>, <math>\mathbf{P}_i(0)</math>, <math>i = 1, \dots, K - 1</math></p> <p>Input: the captured data at time instant <math>t</math></p> <ol style="list-style-type: none"> <li>1) Extract subsystems <math>\mathbf{x}_i(t)</math> as described in section IV-A1</li> <li>2) Tracking the principal subspace of each subsystems using some existing algorithms (e.g. FAPI, OPAST):</li> </ol> $\mathbf{W}_i(t) = \text{TrackingAlgorithm}(\mathbf{W}_i(t-1), \mathbf{x}_i(t)).$ <ol style="list-style-type: none"> <li>3) Recursive update of <math>\mathbf{P}_i(t)</math> and <math>\mathbf{M}(t)^{-1}</math> (Eq. (32) and (33))</li> <li>4) Compute matrix <math>\mathbf{T}_i(t)</math>: <math>\mathbf{T}_i(t) = \mathbf{W}_i^\#(t)(\mathbf{P}_i(t)\mathbf{M}^{-1}(t))\mathbf{W}_1(t)</math>.</li> <li>5) Compute the principal weight matrix <math>\mathbf{W}(t)</math>: <math>\mathbf{W}(t) = [\mathbf{W}_1^T(t), (\mathbf{W}_2^T(t)\mathbf{T}_2^T(t)), \dots, (\mathbf{W}_K^T(t)\mathbf{T}_K^T(t))]^T</math>.</li> </ol> <p>Output: the principal subspace <math>\mathbf{W}(t)</math></p>
---

TABLE VII  
SUMMARY OF GMNS FOR PST: THE CASE OF OVERLAPPING SUBSYSTEMS

<p>Initialize: initialization of the tracking algorithm</p> <p>Input: the captured data at instant time <math>t</math></p> <ol style="list-style-type: none"> <li>1) Extract subsystems <math>\mathbf{x}_i(t)</math> as described in IV-A2</li> <li>2) Tracking the principal subspace of each subsystem using some existing algorithms (e.g. FAPI, OPAST):</li> </ol> $\mathbf{W}_i(t) = \text{TrackingAlgorithm}(\mathbf{W}_i(t-1), \mathbf{x}_i(t)).$ <ol style="list-style-type: none"> <li>3) Extract the overlap between first subsystem and <math>i</math>-th subsystem: <math>\mathbf{W}_1^{\text{olap}}(t)</math> and <math>\mathbf{W}_i^{\text{olap}}(t)</math>.</li> <li>4) Compute the matrix <math>\hat{\mathbf{T}}_i(t)</math>, <math>i = 2, \dots, K</math>: <math>\mathbf{T}_i(t) = (\mathbf{W}_i^{\text{olap}}(t))^\#(\mathbf{W}_1^{\text{olap}}(t))</math>, and the principal weight matrices: <math>\tilde{\mathbf{W}}_i(t) = \mathbf{W}_i(t)\mathbf{T}_i(t)</math> (with <math>\tilde{\mathbf{W}}_1(t) = \mathbf{W}_1(t)</math>).</li> </ol> <p>Output: the principal subspace <math>\mathbf{W}(t)</math></p>
---

Then, the principal eigenvectors are found to be

$$\mathbf{U}_s(t) = \mathbf{W}(t)\tilde{\mathbf{Q}}^{-1}(t). \quad (36)$$

A naive implementation of the above procedure is not numerically efficient. Observed that (35)

TABLE VIII  
SUMMARY OF GMNS FOR PRINCIPAL EIGENVECTOR TRACKING

<p>Input: the principal subspaces <math>\mathbf{W}_k(t)</math>, <math>\mathbf{T}_k(t)</math></p> <ol style="list-style-type: none"> <li>1) Compute <math>\mathbf{D}(t)</math>: <math>\mathbf{D}(t) = [\mathbf{I} + \sum_{k=2}^K \mathbf{T}_k^H(t)\mathbf{T}_k(t)]^{-1}</math>.</li> <li>2) Compute <math>\mathbf{Z}(t)</math> as (44) or (45)</li> <li>3) Compute <math>\tilde{\mathbf{Q}}^{-1}(t)</math>: <math>\tilde{\mathbf{Q}}^{-1}(t) = \text{eig}(\mathbf{Z}(t))</math>.</li> <li>4) Extract <math>p</math> the principal eigenvectors as <math>\mathbf{U}_s(t) = \mathbf{W}(t)\tilde{\mathbf{Q}}^{-1}(t)</math>.</li> </ol> <p>Output: the principal eigenpairs <math>\mathbf{U}_s(t)</math> and <math>\{\lambda_i(t)\}_{i=1:p}</math></p>
--

can be written as

$$\begin{aligned} \mathbf{W}^\#(t)\mathbf{R}_{xx}(t)\mathbf{W}(t) &\cong E\{\mathbf{W}^\#(t)\mathbf{x}(t)\mathbf{x}(t)^H\mathbf{W}(t)\} \\ &\cong E\{\tilde{\mathbf{y}}(t)\mathbf{y}^H(t)\}, \end{aligned}$$

where

$$\tilde{\mathbf{y}}(t) = \mathbf{W}^\#(t)\mathbf{x}(t), \quad (37)$$

$$\mathbf{y}(t) = \mathbf{W}^H(t)\mathbf{x}(t). \quad (38)$$

The main cost thus comes from calculating (37) with the pseudo-inverse operator

$$\begin{aligned} \tilde{\mathbf{y}}(t) &= \mathbf{W}^\#(t)\mathbf{x}(t) \\ &= [\mathbf{W}(t)^H\mathbf{W}(t)]^{-1}\mathbf{W}(t)^H\mathbf{x}(t) \\ &= \mathbf{D}(t)\mathbf{y}(t), \end{aligned}$$

where

$$\begin{aligned} \mathbf{D}(t) &= [\mathbf{W}(t)^H\mathbf{W}(t)]^{-1} \\ &= \left[\mathbf{I} + \sum_{k=2}^K \mathbf{T}_k^H(t)\mathbf{T}_k(t)\right]^{-1}. \end{aligned} \quad (39)$$

Equation (39) comes from the fact that  $\mathbf{W}_k(t)$  for  $k = 1, \dots, K$  are orthogonal (i.e.,  $\mathbf{W}_k^H(t)\mathbf{W}_k(t) = \mathbf{I}$ ). Thus,  $\tilde{\mathbf{y}}(t)$  and  $\mathbf{y}(t)$  can be obtained as

$$\tilde{\mathbf{y}}(t) = \sum_{k=1}^K \tilde{\mathbf{y}}_k(t), \quad (40)$$

$$\mathbf{y}(t) = \sum_{k=1}^K \mathbf{y}_k(t), \quad (41)$$

where

$$\tilde{\mathbf{y}}_k = \mathbf{D}(t)\mathbf{y}_k(t), \quad (42)$$

$$\mathbf{y}_k = \mathbf{W}_k^H(t)\mathbf{x}_k(t), \quad (43)$$

which can be implemented in parallel in each DSP. We then estimate  $\tilde{\mathbf{Q}}^{-1}(t)$  as EVD of  $\mathbf{Z}(t)$ , which is obtained as follows. If sliding window is used, then

$$\begin{aligned} \mathbf{Z}(t) &= \sum_{\tau=0}^{L-1} \tilde{\mathbf{y}}(t-\tau)\mathbf{y}^H(t-\tau) \\ &= \mathbf{Z}(t-1) + \tilde{\mathbf{y}}(t)\mathbf{y}^H(t) - \tilde{\mathbf{y}}(t-L+1)\mathbf{y}^H(t-L+1), \end{aligned} \quad (44)$$

where  $L$  is the sliding window size. Otherwise,

$$\begin{aligned} \mathbf{Z}(t) &= \sum_{\tau=0}^t \beta^{t-\tau} \tilde{\mathbf{y}}(\tau)\mathbf{y}^H(\tau) \\ &= \beta\mathbf{Z}(t-1) + \tilde{\mathbf{y}}(t)\mathbf{y}^H(t), \end{aligned} \quad (45)$$

if the exponential window is used ( $0 < \beta \leq 1$  being the forgetting factor). Once  $\tilde{\mathbf{Q}}^{-1}(t)$  has been estimated, the principal eigenvectors are obtained as given by (36), which can again be estimated in parallel. The algorithm costs  $O(np^2/K) + O(p^3)$  and is summarized in Table VIII.

## VI. ASYMPTOTIC PERFORMANCE ANALYSIS

In this section, we derive the result of the first order asymptotic performance analysis of the standard subspace estimation method. Derivation details can be found in Appendix B. Here, we summarize the two key steps and results <sup>6</sup>.

First, calculate the asymptotic covariance of each error eigenvectors as

$$E[\delta\mathbf{u}_i\delta\mathbf{u}_j^T] = \frac{\lambda_i}{T} \sum_{\substack{k=1 \\ k \neq i}}^n \frac{\lambda_k}{(\lambda_k - \lambda_i)^2} \mathbf{u}_k \mathbf{u}_k^T \delta_{ij} \quad (46)$$

<sup>6</sup>For simplicity, we present the results in the real case.

where  $\delta_{ij}$  refers to the Kronecker index, and  $\mathbf{u}_i$  and  $\hat{\mathbf{u}}_i$  are the eigenvector and the perturbed eigenvector of the covariance matrix  $\mathbf{R}_{xx}$  and the sample covariance matrix  $\hat{\mathbf{R}}_{xx}$ , respectively. Thus, the error vector is  $\delta\mathbf{u}_i = \hat{\mathbf{u}}_i - \mathbf{u}_i$ . Note that in (46), we assumed that  $\lambda_i \neq \lambda_j$ , for all  $i, j = 1, \dots, p$ .

Second, calculate covariance of the perturbed signal subspace projection matrix  $\boldsymbol{\pi} = \sum_{i=1}^p \mathbf{u}_i \mathbf{u}_i^T$  as follows:

$$\begin{aligned} & E[\text{vec}(\delta\boldsymbol{\pi}) \text{vec}(\delta\boldsymbol{\pi})^T] \\ &= \sum_{i=1}^p \nabla g(\mathbf{u}_i) E\{\delta\mathbf{u}_i \delta\mathbf{u}_i^T\} \nabla g(\mathbf{u}_i)^T, \end{aligned} \quad (47)$$

where  $\text{vec}(\cdot)$  denotes the matrix vectorization operator, and

$$\hat{\boldsymbol{\pi}} = \sum_{i=1}^p \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i^T, \quad (48)$$

$$\delta\boldsymbol{\pi} = \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}, \quad (49)$$

$$\nabla g(\mathbf{u}_i) = (\mathbf{u}_i \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{u}_i). \quad (50)$$

A similar derivation of the asymptotic covariance of GMNS-PSA method has been conducted but is omitted here due to its inefficiency as explained next. Unfortunately, this performance evaluation approach is inefficient in our context.

Indeed, as showed by simulation later, the curve of the experimental method (i.e., empirical error estimation) is always lower than the curve of the analytical method (i.e., theoretical error estimation in (47)). It means that the first-order method is not good enough and needs to be replaced by the second-order one<sup>7</sup>.

For the standard method, we can easily develop a second-order asymptotic performance analysis by directly applying the results in [23] instead of using (46). However, this approach is tedious and formulas are extremely long and complex. For this reason, the performance evaluation presented in this paper given in Section VII is based on simulated experiments.

<sup>7</sup>This problem has been observed in other contexts, e.g. [22], and is due to the fact that the estimation error is quite small and hence the asymptotic would require hundred of thousands of observations to be reached.

## VII. SIMULATIONS

In this section, the performance of the subspace estimation is verified by simulation. In all experiments, the system matrix  $\mathbf{A}$  is randomly generated but kept fixed for all Monte Carlo runs. The sources are i.i.d. Gaussian processes of unit power. The noise is spatially white with specific SNRs defined as

$$\text{SNR} = 10 \log_{10} \frac{\|\mathbf{A}\|^2}{\sigma^2}, \quad (51)$$

where  $\sigma^2$  is the noise variance.

To assess the performance of the proposed algorithms, the following two criteria are used. The first one is the subspace estimation performance (SEP) defined as

$$\text{SEP}(t) = \frac{1}{R} \sum_{i=1}^R \frac{\text{Tr}\{\mathbf{W}_i^H(t)(\mathbf{I} - \mathbf{W}_{\text{ex}}(t)\mathbf{W}_{\text{ex}}^H(t))\mathbf{W}_i(t)\}}{\text{Tr}\{\mathbf{W}_i^H(t)(\mathbf{W}_{\text{ex}}(t)\mathbf{W}_{\text{ex}}^H(t))\mathbf{W}_i(t)\}}, \quad (52)$$

where  $R$  is the number of Monte Carlo run,  $\mathbf{W}_i$  is the estimated subspace at  $i$ -th run, and  $\mathbf{W}_{\text{ex}}$  is the exact subspace weight matrix computed by orthogonalizing  $\mathbf{A}$ . In batch system case, we can drop  $t$  from SEP.

The second criterion is the eigenvectors estimation performance (EEP) defined as

$$\text{EEP}(i) = \frac{1}{R} \sum_{i=1}^R \|\mathbf{U}_i - \mathbf{U}_{\text{ex}}\|_F^2, \quad (53)$$

where  $\mathbf{U}_i$  is the matrix of the estimated eigenvectors<sup>8</sup>, and  $\mathbf{U}_{\text{ex}}$  is the matrix of exact eigenvectors computed from the exact covariance matrix (i.e. noiseless case) using full SVD algorithm.

In all cases,  $R$  is set to 100. To evaluate performance of the algorithms with respect to the number of sources,  $p$ , we present two different scenarios ( $p = 2$  and  $p = 6$ ) for relatively large dimensional systems. A summary of parameters used in the experiments is given in Table IX.

### A. Minor Subspace Analysis

First, we assess the performance of the GMNS-MSA against the standard method (SVD-MSA) with both small ( $p = 2$ ) and large ( $p = 6$ ) numbers of sources. The results indicate that the GMNS-MSA has performance close to the standard method (SVD-MSA) when  $p = 2$ , as shown in Figure 5, and it lost some accuracy at low SNRs (i.e.,  $\text{SNR} < 5$  dB) when  $p = 6$ , as shown in Figure 6. However, in both experiments, the dominant cost of the GMNS-MSA is reduced by a factor of  $K^2$ , as compared to the SVD. The selected GPCS is given in Table X.

<sup>8</sup>To remove inherent ambiguities, the eigenvector norm is set to 1 and its first entry is chosen to be positive real-valued.

TABLE IX  
PARAMETERS USED IN OUR EXPERIMENTS.

Experiment	Figure	$n$	$p$	$K$	$T$	$q$
1	5	26	2	4	500	
2	6	26	6	4	500	
3	7	40	2	4	500	4
4	8	40	6	4	500	8
5	9	40	2	4	500	4
6	10	40	6	4	500	8
7	11	26	2	4	4000	
8	12	26	6	4	4000	
9	13	40	2	4	3000	4
10	14	40	6	4	3000	8
11	15	40	2	4	3000	4
12	16	40	6	4	3000	8

TABLE X  
GPCS USED FOR FIRST EXPERIMENT.

Tuples	Members
$t_1$	$(m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8)$
$t_2$	$(m_1, m_2, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14})$
$t_3$	$(m_1, m_2, m_{15}, m_{16}, m_{17}, m_{18}, m_{19}, m_{20})$
$t_4$	$(m_1, m_2, m_{21}, m_{22}, m_{23}, m_{24}, m_{25}, m_{26})$

### B. Principal Subspace/Component Analysis

For principal subspace analysis, we compare the performance of the three proposed algorithms – GMNS-N-PSA, GMNS-O-PSA and GMNS-NU-PSA – against the standard principal subspace analysis method (SVD-PSA), for  $p = 2$  and  $p = 6$  as shown in Figures 7 and 8, respectively. Among the four algorithms, GMNS-N-PSA has the lowest performance, as noticed in Section IV-A1. The three other methods reach the SVD, except at low SNRs.

For principal component analysis, the same observation is done as for PSA, as shown in Figures 9 and 10. Similar to the MSA case, our GMNS based algorithms have the advantage of



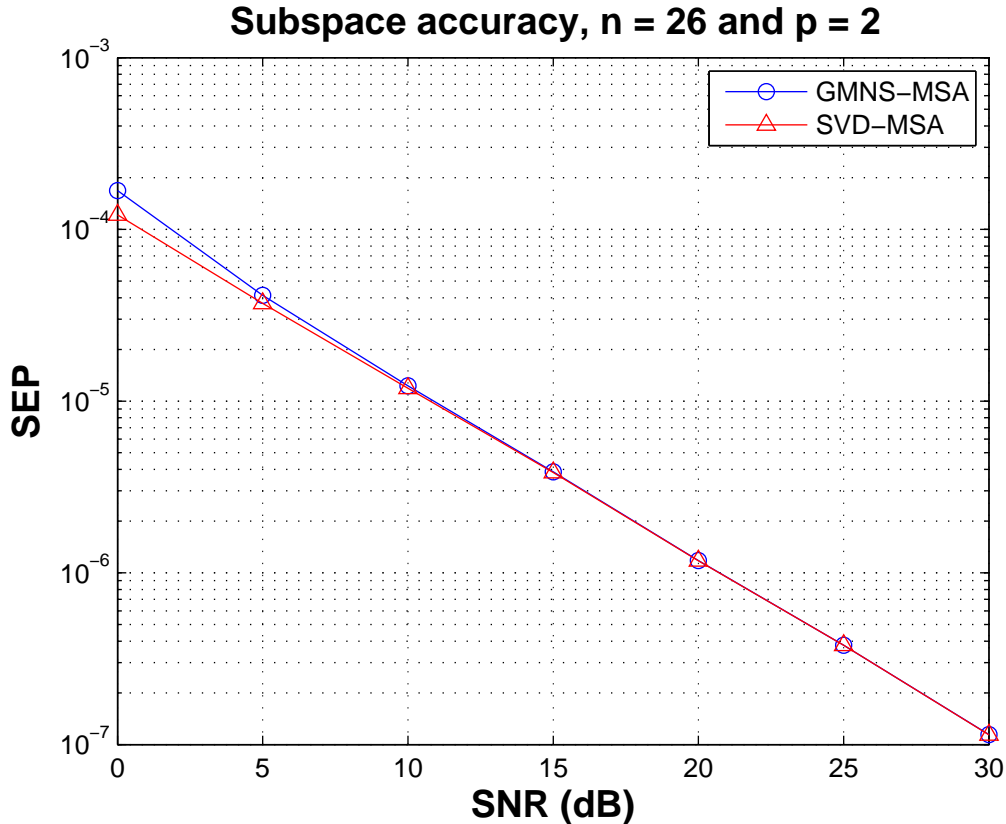


Fig. 5. Minor subspace estimation ( $n = 26, p = 2$ ).

lower computational cost with a gain approximately equal to  $K^2$ .

### C. Minor Subspace Tracking

For minor subspace tracking, we chose two low cost algorithms, FOOja [20] and FDPM [11], and one moderate cost one, YAST [24], and then compare their performance with the corresponding GMNS-based algorithms: GMNS-MST-FOOja, GMNS-MST-FDPM and GMNS-MST-YAST. All algorithms run under noisy environment at SNR = 15 dB. The performance results with respect to  $p = 2$  and  $p = 6$  are shown in Figures 11 and 12, respectively.

Interestingly, the performance of the GMNS-MST-FOOja is better than the original FOOja even though its convergence rate is slower. The reason is that dividing data into small subsystems reduces the search space which mitigates the local minima convergence problem and enhances the overall performance. An analogous observation can be seen in the FDPM algorithm. A better

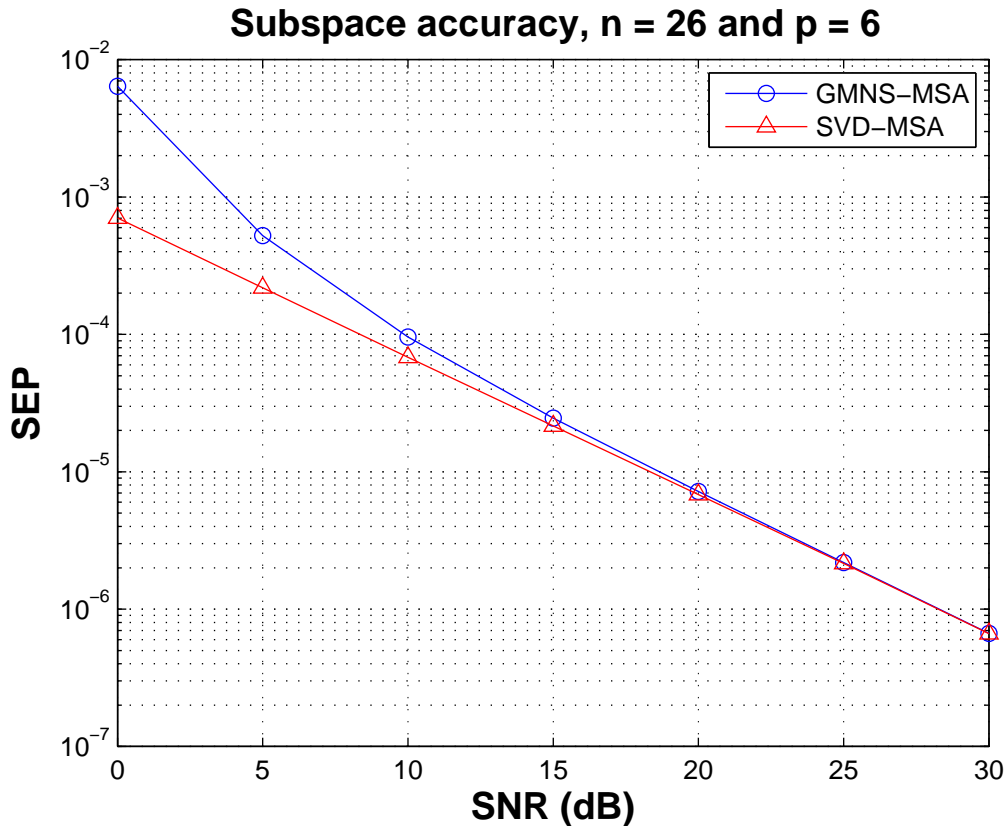


Fig. 6. Minor subspace estimation ( $n = 26, p = 6$ ).

convergence rate and estimated accuracy are obtained by YAST<sup>9</sup> and GMNS-MST-YAST but at the cost of a higher computational complexity.

#### D. Principal Subspace Tracking

For principal subspace tracking, we compare the OPAST and FDPM algorithms<sup>10</sup>, with their corresponding GMNS-based algorithms: GMNS-N-PST-OPAST, GMNS-O-PST-OPAST, GMNS-N-PST-FDPM and GMNS-O-PST-FDPM. It can be seen from Figures 13 and 14 that the GMNS-based algorithms have the same performance as their original algorithms, but with a reduced cost. Also, we observe a clear advantage in favor of OPAST-based methods as compared to FDPM-based ones.

<sup>9</sup>YAST has complexity of  $O(np^2)$  as presented in [24].

<sup>10</sup>Principal subspace can be obtained from FDPM by changing the sign of step size parameter.

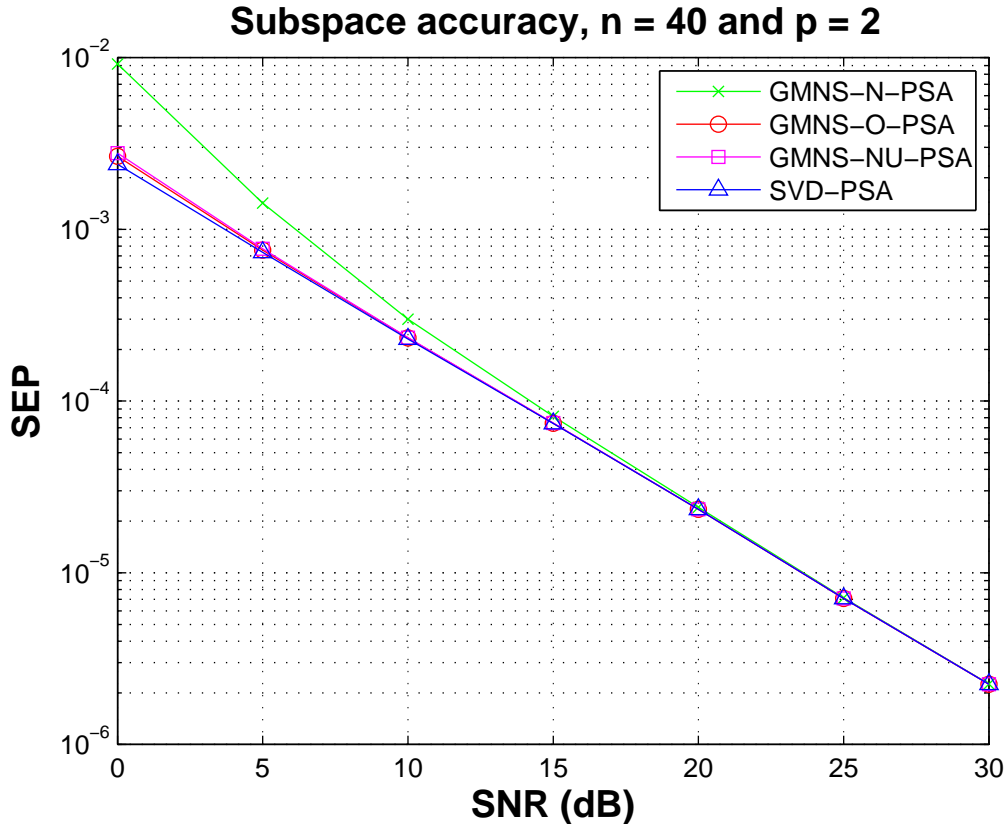


Fig. 7. Principal subspace estimation ( $n = 40, p = 2$ ).

### E. Principal Eigenvector Tracking

As showed in the previous section, the performance of the GMNS-based FDPM is worse than the GMNS-based OPAST. Thus, we apply the procedure in Section V-C to the parallelized GMNS-based OPAST algorithms (GMNS-N-PST-OPAST and GMNS-O-PST-OPAST) to obtain the principal eigenvectors. Then, we compare their results with the eigenvectors estimation achieved by the SVD algorithm. As showed in Figures 15 and 16, both GMNS-N-PST-OPAST and GMNS-O-PST-OPAST algorithms have close performance as compared to the SVD-PST.

### F. Performance Analysis

In this section, we use Monte-Carlo simulation to assess the analytical performance as given in Equation (47) in the standard subspace estimation case. As a performance criterion, we use the trace of the estimation error covariance of the signal subspace projection matrix, that is,

$$\varrho = \text{Tr}(E[\text{vec}(\delta\boldsymbol{\pi}) \text{vec}(\delta\boldsymbol{\pi})^T]). \quad (54)$$

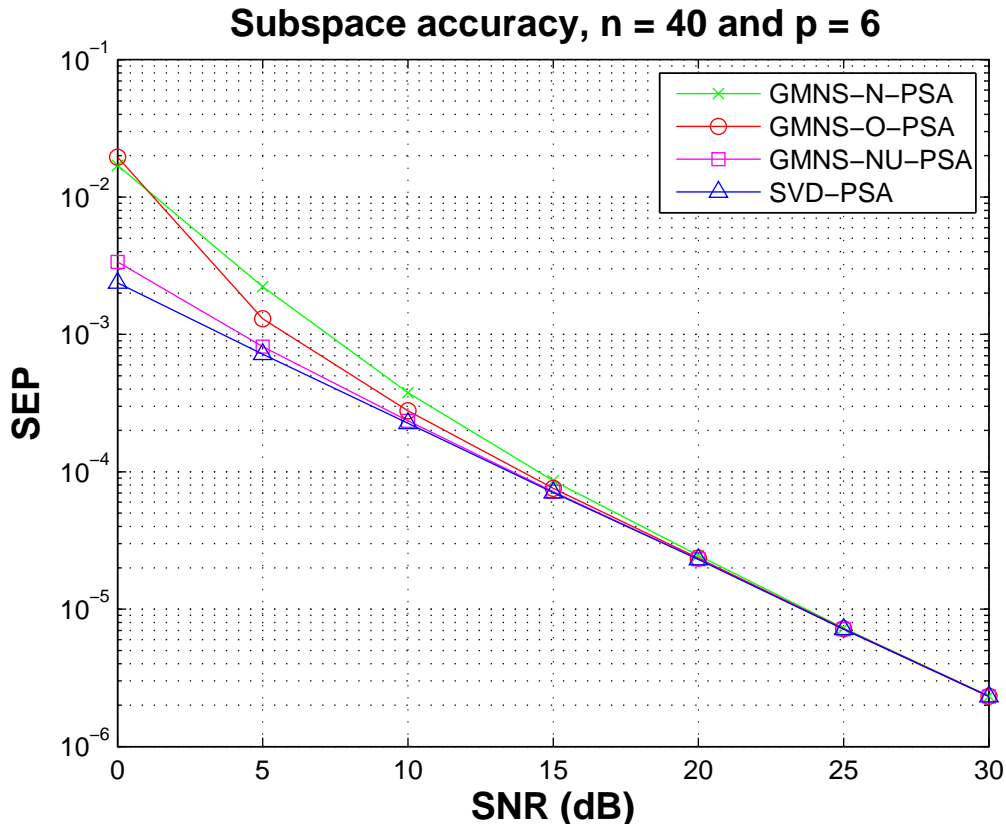


Fig. 8. Principal subspace estimation ( $n = 40, p = 6$ ).

This criterion is drawn as function of time index.

From Figure 17 we can see that the experimental curve is much lower than the analytical one. This is also true for all other experiments we have run. Thus, we conclude that the first-order asymptotic performance is not accurate in this case. A second-order one may help but such an analysis is out of the scope of this paper. Note that a similar observation was made in [22].

## VIII. CONCLUSIONS

In this paper, we have proposed a simple but efficient approach for the estimation and tracking of the signal and noise subspaces. The considered problems are quite common in many array processing applications and are known as the most expensive tasks in source localization and source separation problems to the extent that many efficient spatial filtering methods have been disregarded in real-life applications which use large antenna arrays, e.g., radiofrequency inter-

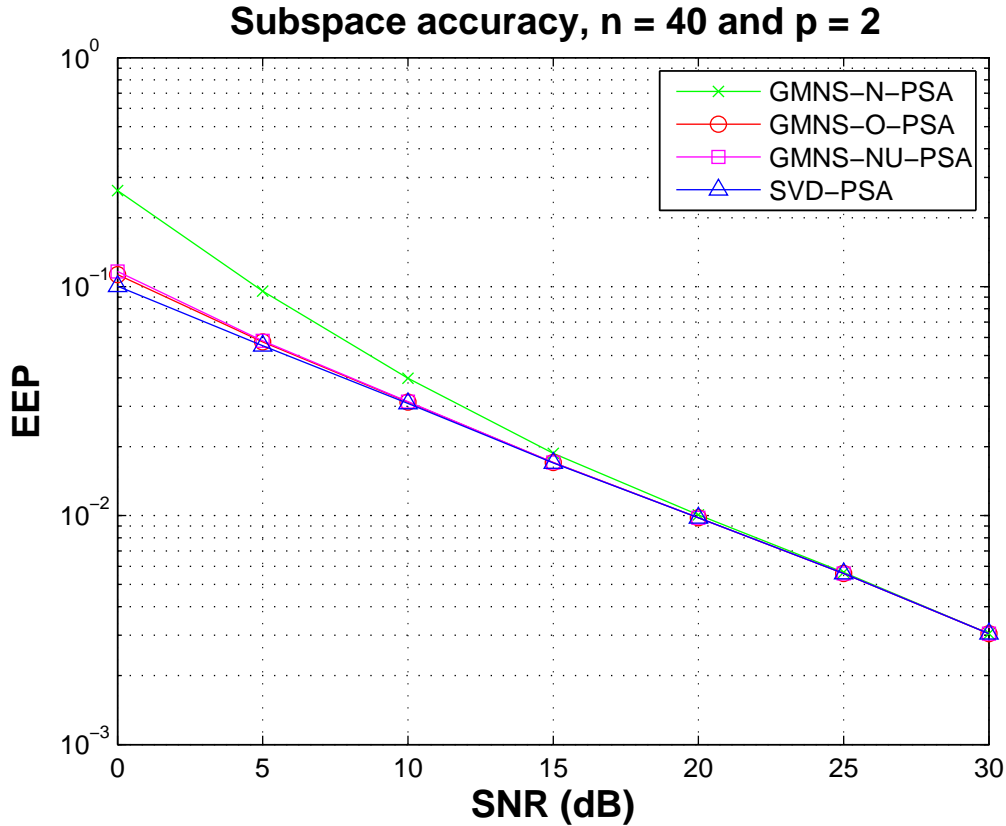


Fig. 9. Principal eigenvector estimation ( $n = 40, p = 2$ ).

ference (RFI) mitigation in radioastronomy [25], [26]<sup>11</sup>. Our solution (GMNS method) proposes to exploit the specific array processing model together with a parallel computation architecture to reduce the overall cost by a factor close to  $K^2$  – the number of parallel computational units – for large dimensional systems. Several algorithmic versions of the GMNS have been developed and the performance was assessed via simulated experiments. The performance results showed clearly that the GMNS represents an excellent solution to deal with large size arrays when distributed resources or several parallel computational units are available.

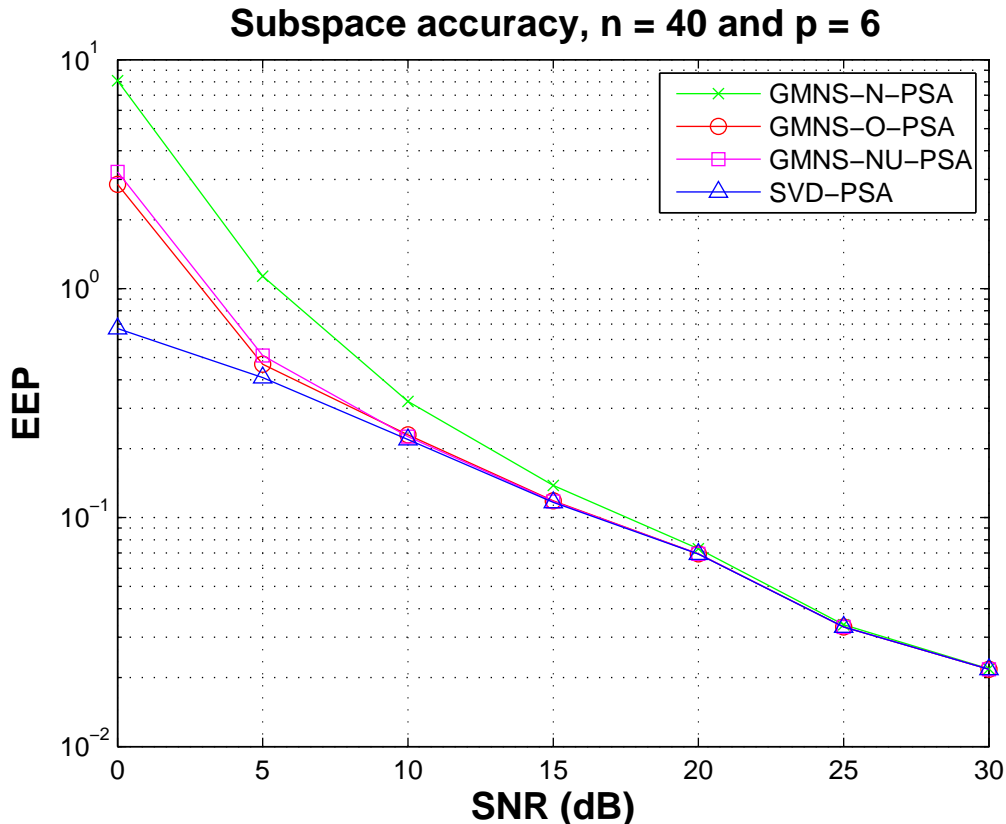


Fig. 10. Principal eigenvector estimation ( $n = 40, p = 6$ ).

## APPENDIX A

### PROOF OF THEOREM 1

To prove Theorem 1, we need to show that the columns of  $\mathbf{V}_i$  belong to the noise subspace and the columns of  $\mathbf{V}$  form a vector basis of noise subspace.

First, note that  $\tilde{\mathbf{V}}_i^H \mathbf{A}_i = \mathbf{0}$  leads to  $\mathbf{V}_i^H \mathbf{A} = \mathbf{0}$  because of the zero padding procedure. Hence the columns of  $\mathbf{V}_i$  belong to the noise subspace.

To prove that the columns of  $\mathbf{V}$  form a vector basis, let show that the noise matrix has (up to row permutation) a block diagonal structure as illustrated in Figure 18 with non-singular  $d \times d$  diagonal blocks, which guarantee its full column rank. Indeed, according to the GPCS concept, the  $i$ -th  $d \times d$  diagonal block represents the entries of  $\mathbf{V}_i$  corresponding to the  $d$  system outputs not shared by the preceding subsystems (i.e., associated to tuples  $1, 2, \dots, i - 1$ ). The block

<sup>11</sup>Due to their high computational cost, efficient subspace based RFI mitigation methods are replaced in practice by a simple but efficient RFI or 'No RFI' labeling RFI method.

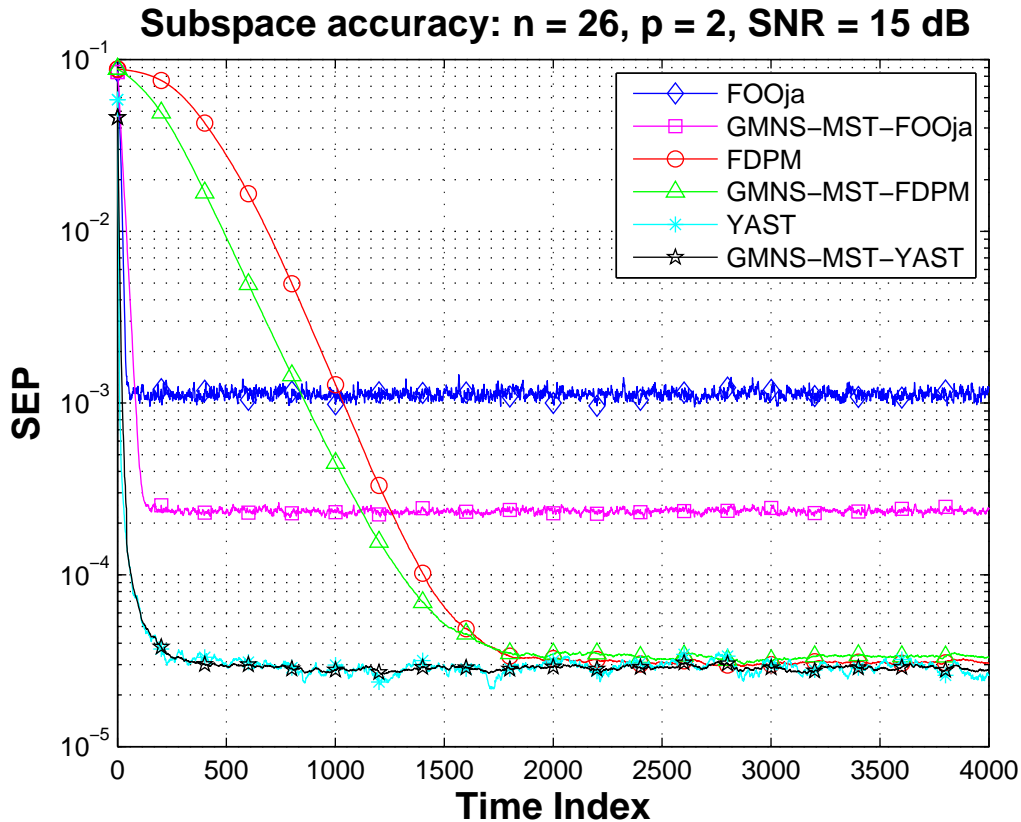


Fig. 11. Minor subspace tracking performance ( $n = 26, p = 2$ ).

diagonal structure is then a direct consequence of the zero padding technique used to build  $\mathbf{V}_i$  from  $\tilde{\mathbf{V}}_i$ .

Let us prove now that the  $d \times d$  diagonal blocks are non-singular. Indeed, if a given diagonal block matrix is singular, then there exists a noise vector with at most  $p$  non-zero entries which is in contradiction with the assumption that any  $p$  rows of matrix  $\mathbf{A}$  are linearly independent, as assumed in Theorem 1.

## APPENDIX B

### DERIVATION OF FIRST-ORDER ASYMPTOTIC PERFORMANCE ANALYSIS

The main idea to derive the asymptotic performance analysis is that, at each step, we write the results as a function of asymptotic covariance and keep only first order terms, as presented next.

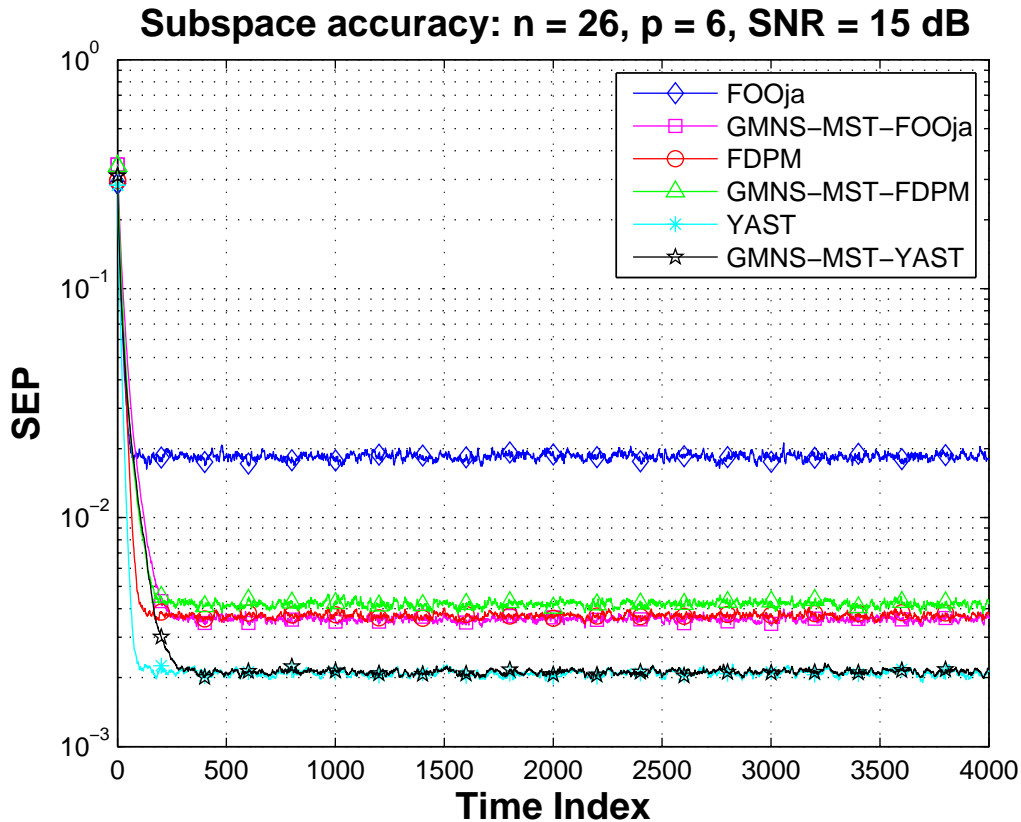


Fig. 12. Minor subspace tracking performance ( $n = 26, p = 6$ ).

To make the notation consistent, the perturbed matrices or vectors are distinguished with their true ones by the hat mark. For example, the sample covariance matrix is defined as  $\hat{\mathbf{R}} = \sum_t \mathbf{x}(t)\mathbf{x}^H(t)$  while the true one is  $\mathbf{R} = E[\mathbf{x}(t)\mathbf{x}^H(t)]$ . Now, we provide a step-by-step derivation of the asymptotic covariance of the standard method.

Let  $\mathbf{u}_1, \dots, \mathbf{u}_n$  and  $\lambda_1, \dots, \lambda_n$  be the eigenvectors and the eigenvalues of  $\mathbf{R}$ , respectively. By assuming that the eigenvalues are distinct, it leads to

$$\hat{\mathbf{R}}\hat{\mathbf{u}}_i = \hat{\lambda}_i\hat{\mathbf{u}}_i, \quad (55)$$

where  $\hat{\mathbf{R}} = \mathbf{R} + \delta\mathbf{R}$ ,  $\hat{\mathbf{u}}_i = \mathbf{u}_i + \delta\mathbf{u}_i$ , and  $\hat{\lambda}_i = \lambda_i + \delta\lambda_i$ . We keep the first-order term of (55) and, after some manipulations, then obtain

$$\begin{aligned} \delta\mathbf{u}_i &= -\mathbf{\Gamma}^\# \delta\mathbf{R}\mathbf{u}_i \\ &= \sum_{k \neq i} \frac{\mathbf{u}_k}{\lambda_k - \lambda_i} \mathbf{u}_k^T \delta\mathbf{R}\mathbf{u}_i, \end{aligned}$$



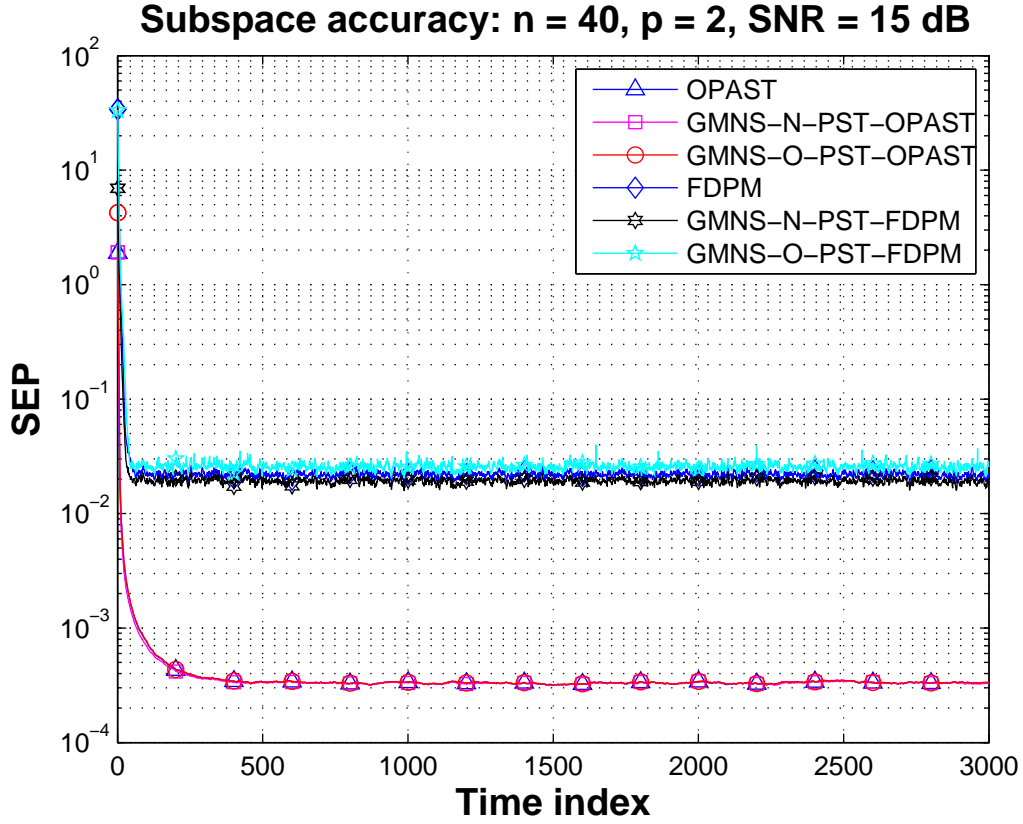


Fig. 13. Principal subspace tracking performance ( $n = 40, p = 2$ ).

where

$$\begin{aligned} \Gamma &= \mathbf{R} - \lambda_i \mathbf{I} \\ &= \sum_{\substack{k=1 \\ k \neq i}}^n (\lambda_k - \lambda_i) \mathbf{u}_k \mathbf{u}_k^T. \end{aligned}$$

Hence,

$$\begin{aligned} E[\delta \mathbf{u}_i \delta \mathbf{u}_j^T] &= E \left[ \sum_{\substack{k \neq i \\ l \neq j}} \frac{\mathbf{u}_k}{\lambda_k - \lambda_i} (\mathbf{u}_k^T \delta \mathbf{R} \mathbf{u}_i) (\mathbf{u}_l^T \delta \mathbf{R} \mathbf{u}_j) \frac{\mathbf{u}_l^T}{\lambda_l - \lambda_j} \right] \\ &= \sum_{\substack{k \neq i \\ l \neq j}} \frac{\mathbf{u}_k}{\lambda_k - \lambda_i} E \{ (\mathbf{u}_k^T \delta \mathbf{R} \mathbf{u}_i) (\mathbf{u}_l^T \delta \mathbf{R} \mathbf{u}_j) \} \frac{\mathbf{u}_l^T}{\lambda_l - \lambda_j}. \end{aligned} \quad (56)$$

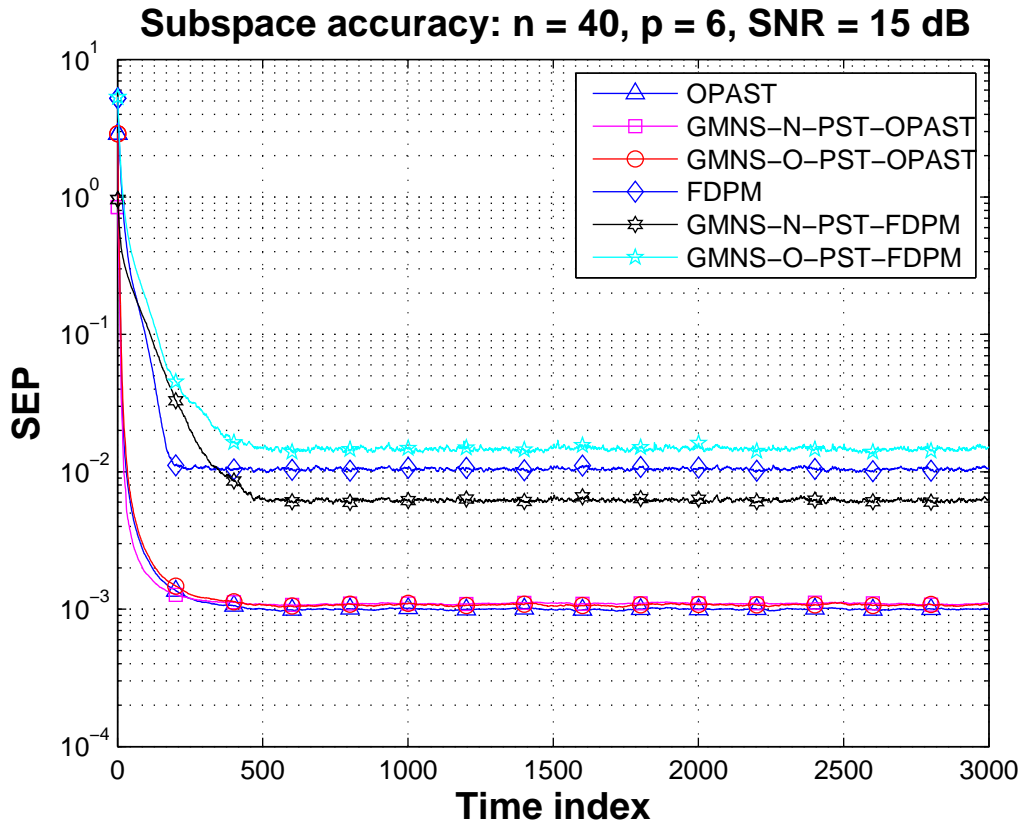


Fig. 14. Principal subspace tracking performance ( $n = 40, p = 6$ ).

Before handling further, we note that [27]

$$E[(\mathbf{u}_k^T \delta \mathbf{R} \mathbf{u}_i)(\mathbf{u}_j^T \delta \mathbf{R} \mathbf{u}_l)] = \frac{1}{T} (\mathbf{u}_k^T \mathbf{R} \mathbf{u}_l)(\mathbf{u}_j^T \mathbf{R} \mathbf{u}_i). \quad (57)$$

By combining (56) and (57), we achieve (46).

Our next objective is to obtain (47). As defined in (48), we have

$$\text{vec}(\boldsymbol{\pi}) = \sum_{i=1}^p \mathbf{u}_i \otimes \mathbf{u}_i = \sum_{i=1}^p g(\mathbf{u}_i),$$

where  $g(\mathbf{u}_i) = \mathbf{u}_i \otimes \mathbf{u}_i$ . Taking the first-order Taylor expansion yields

$$g(\hat{\mathbf{u}}_i) \simeq g(\mathbf{u}_i) + \nabla g(\mathbf{u}_i)(\hat{\mathbf{u}}_i - \mathbf{u}_i),$$

where  $\nabla$  refers to the gradient given in (50). We can rewrite now

$$\text{vec}(\delta \boldsymbol{\pi}) = \sum_{i=1}^p \nabla g(\mathbf{u}_i) \delta \mathbf{u}_i. \quad (58)$$

By taking the covariance of  $\text{vec}(\delta \boldsymbol{\pi})$  and using (46), we obtain the final expression of (47).

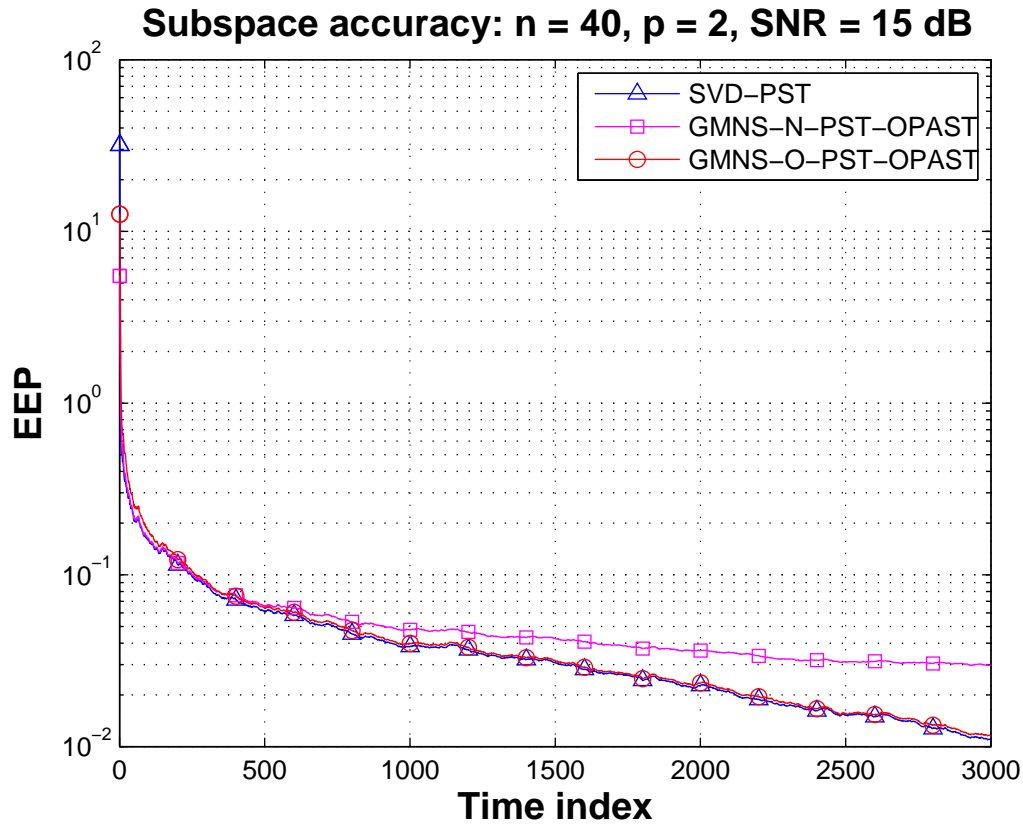


Fig. 15. Principal eigenvector estimation ( $n = 40, p = 2$ ).

#### ACKNOWLEDGMENT

The authors would like to acknowledge the support from the Grant 102.02-2015.32 of the Vietnam National Foundation for Science and Technology Development.

#### REFERENCES

- [1] P. Stoica, R. Moses, *Spectral Analysis of Signals*. Prentice Hall, 2005.
- [2] P. Comon and Ch. Jutten, *Handbook of Blind Source Separation: ICA and Applications*. Elsevier, 2010.
- [3] G. Hellboug, “RFI spatial processing for modern radio telescopes,” *PhD Thesis, Univ. of Orléans*, Jan. 2014.
- [4] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.
- [5] S. Haykin, K. J. Ray Liu, *Handbook on Array Processing and Sensor Networks*. Wiley-IEEE Press, 2010.
- [6] E. G. Larsson, F. Tufvesson, O. Edfors, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, 2013.
- [7] “Square kilometre array: Exploring the universe with the world’s largest radio telescope,” <https://www.skatelescope.org/>.
- [8] A. Bertrand and M. Moonen, “Distributed adaptive eigenvector estimation of the sensor signal covariance matrix in a fully connected sensor network,” in *Proc. ICASSP*, May 2013, pp. 4236–4240.

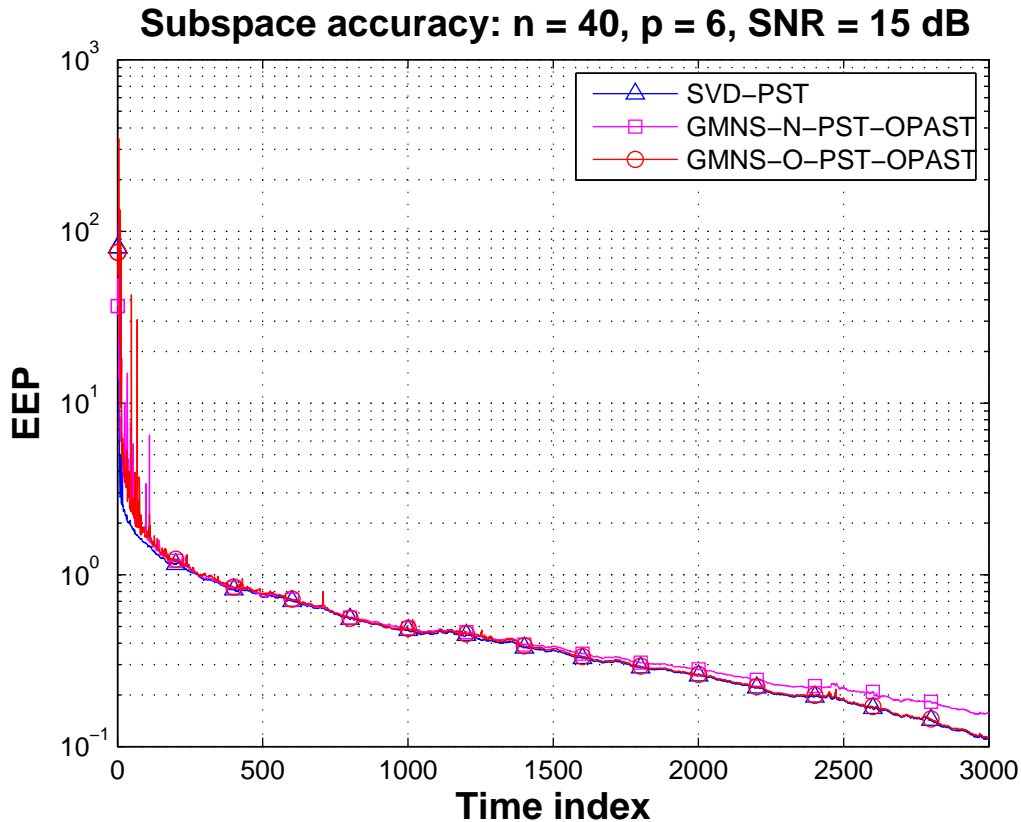


Fig. 16. Principal eigenvector estimation ( $n = 40$ ,  $p = 6$ ).

- [9] B. Yang, "Projection approximation subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 43, no. 1, pp. 95–107, 1995.
- [10] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal past algorithm," *Signal Processing Letters, IEEE*, vol. 7, no. 3, pp. 60–62, 2000.
- [11] X. G. Doukopoulos and G. V. Moustakides, "Fast and stable subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 56, no. 4, pp. 1452–1465, 2008.
- [12] K. Abed-Meraim and Y. Hua, "Blind identification of multi-input multi-output system using minimum noise subspace," *IEEE Tr-SP*, no. 1, pp. 254–258, 1997.
- [13] K. Abed-Meraim, Y. Hua and A. Belouchrani, "Minimum noise subspace: concepts and applications," in *Proc. ICICS*, 1997, pp. 118–121.
- [14] W. Soudene, K. Abed-Meraim, and A. Beghdadi, "A new look to multichannel blind image deconvolution," *Image Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1487–1500, 2009.
- [15] M. Thameri, K. Abed-Meraim and A. Belouchrani, "Minor subspace tracking using MNS technique," in *Proc. ICASSP*, 2012, pp. 2433–2436.
- [16] V.-D. Nguyen, K. Abed-Meraim, N. Linh-Trung, and R. Weber, "Generalized MNS method for parallel minor and principal subspace analysis," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*. IEEE, 2014, pp. 2265–2269.

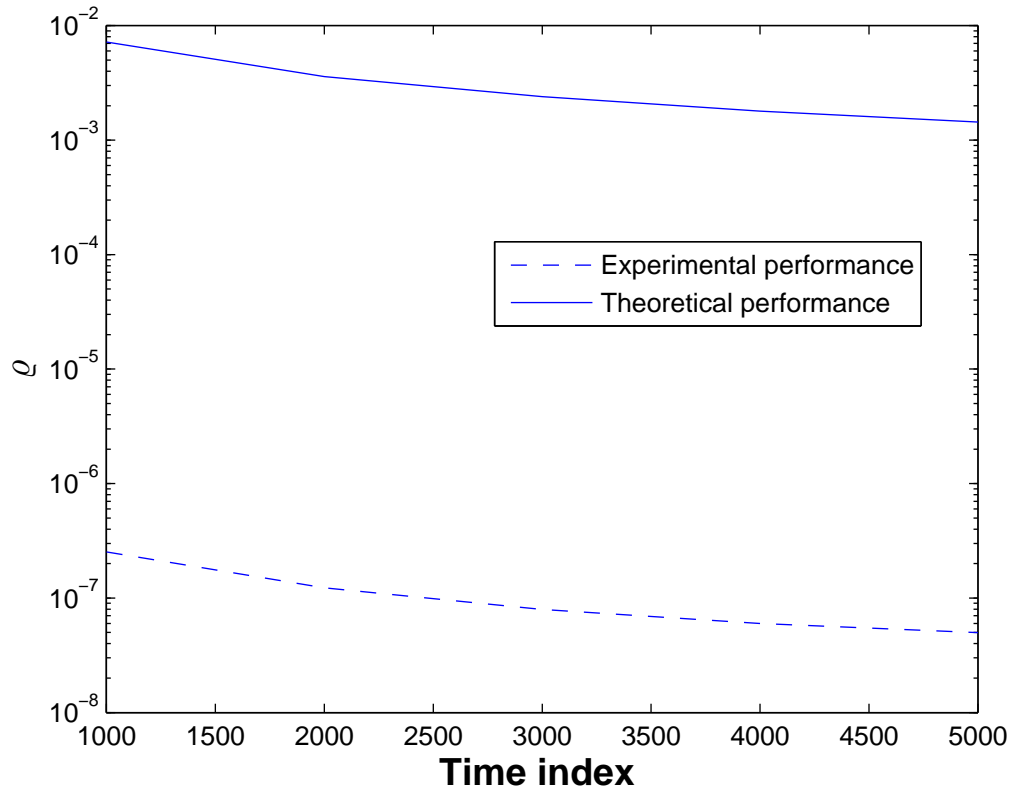


Fig. 17. Performance analysis ( $n = 26, p = 2$ ).

- [17] M. Thameri, K. Abed-Meraim, and A. Belouchrani, "Minor subspace tracking using mns technique," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 2433–2436.

$$\mathbf{V} = \begin{pmatrix}
 \overbrace{\begin{matrix} p \\ \square \\ d \end{matrix}}^d & & & & \square \\
 & \dots & & & \mathbf{X} \\
 & & \ddots & & \vdots \\
 & & & \square & \vdots \\
 & & & & \ddots \\
 \mathbf{0} & & & & \square
 \end{pmatrix}$$

Fig. 18. Block diagonal structure of matrix  $\mathbf{V}$ .

- [18] G. Xu and T. Kailath, "Fast subspace decomposition," *Signal Processing, IEEE Transactions on*, vol. 42, no. 3, pp. 539–551, 1994.
- [19] G.H. Golub, C. F. Van Loan, *Matrix computations*. JHU Press, 2012.
- [20] S. Bartelmaos and K. Abed-Meraim, "Principal and minor subspace tracking: Algorithms and stability analysis," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3, May 2006, pp. III–III.
- [21] R. Badeau, K. Abed-Meraim, G. Richard, and B. David, "Sliding window orthonormal past algorithm," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2003, pp. 261–264.
- [22] K. Abed-Meraim, J.-F. Cardoso, A. Y. Gorokhov, P. Loubaton, and E. Moulines, "On subspace methods for blind identification of single-input multiple-output fir systems," *Signal Processing, IEEE Transactions on*, vol. 45, no. 1, pp. 42–55, 1997.
- [23] Z. Xu, "Perturbation analysis for subspace decomposition with applications in subspace-based algorithms," *Signal Processing, IEEE Transactions on*, vol. 50, no. 11, pp. 2820–2830, 2002.
- [24] R. Badeau, G. Richard, and B. David, "Fast and stable yast algorithm for principal and minor subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3437–3446, 2008.
- [25] G. Hellbourg, R. Weber, K. Abed-Meraim, and A. Boonstra, "RFI spatial processing at nancay observatory: Approaches and experiments," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5387–5391.
- [26] R. Weber, G. Hellbourg, C. Dumez-Viou, A. Boonstra, S. Torchinsky, C. Capdessus, and K. Abed-Meraim, "RFI mitigation in radio astronomy: an overview," in *Les Journées Scientifiques d'URSI-France L'électromagnétisme*, 2013.
- [27] B. Friedlander and A. J. Weiss, "On the second-order statistics of the eigenvectors of sample covariance matrices," *Signal Processing, IEEE Transactions on*, vol. 46, no. 11, pp. 3136–3139, 1998.