Asymptotic Performance Analysis of Subspace Adaptive Algorithms Introduced in the Neural Network Literature

Jean-Pierre Delmas, Florence Alberge

▶ **To cite this version:**

Jean-Pierre Delmas, Florence Alberge. Asymptotic Performance Analysis of Subspace Adaptive Algorithms Introduced in the Neural Network Literature. IEEE Transactions on Signal Processing, 1998. hal-03435761

HAL Id: hal-03435761
https://hal.science/hal-03435761

Submitted on 18 Nov 2021

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.

Asymptotic Performance Analysis of Subspace Adaptive Algorithms Introduced in the Neural Network Literature.

Jean-Pierre Delmas [*] Florence Alberge [†]

Abstract

In the neural network literature, many algorithms have been proposed for estimating the eigenstructure of covariance matrices. We first show that many of these algorithms, when presented in a common framework, show great similitudes with the gradient-like stochastic algorithms usually encountered in the signal processing literature. We derive the asymptotic distribution of these different recursive subspace estimators. A closed-form expression of the covariances in distribution of eigenvectors and associated projection matrix estimators are given and analyzed. In particular, closed-form expressions of the mean square error of these estimators are given. It is found that these covariance matrices have a structure very similar to those describing batch estimation techniques. The accuracy of our asymptotic analysis is checked by numerical simulations, and it is found to be valid not only for a "small" step size but in a very large domain. Finally, convergence speed and deviation from orthonormality of the different algorithms are compared and several tradeoffs are analyzed.

1 Introduction

Over the past decade, adaptive estimation of subspaces of covariance matrices has been applied successfully in signal processing to high resolution spectral analysis and, more recently, to the so-called subspace

[*]Institut National des Télécommunications. 9 rue Charles Fourier, 91011 Evry Cedex, France. tel: (33)-1-60 76 46 32. fax: (33)-1-60 76 42 84. Email: delmas@int-evry.fr

[†]Ecole Nationale Supérieure des Télécommunications. 46 rue Barrault, 75634 Paris Cedex, France. tel: (33)-1-45 81 77 82. fax: (33)-1-45 88 79 35. Email: alberge@sig.enst.fr

approach that is used in blind identification of multichannel FIR filters [1]. At the same time, many neural network realizations have been proposed for the statistical technique of principal component analysis in data compression and feature extraction as well as for optimal fitting in the total least square sense. Among these realizations, several stochastic approximation gradient-like algorithms were proposed by authors in the neural network community. These algorithms have been studied from two points of view only: on the one hand, their neural implementation, and, on the other hand, their convergence analysis in a decreasing step size situation, using the stability study of the associated ordinary differential equation (ODE), see [2], [3] and the references therein. A classic paper on the practical numerical algorithms is [4]. In a constant step size situation, it has been shown [5] that the sequence of estimates can be approximated by the associated ODE, in the sense of weak convergence of random processes as the step size tends to zero. However, the analysis of their asymptotic performance has not yet been studied. The purpose of this paper is to use the approach developed in [6], [7], and [8] to study the more common adaptive algorithms introduced in the neural network literature.

This paper is organized as follows. In Section 2, we give an overview of the main subspace adaptive algorithms introduced in the neural network literature. These algorithms are presented in a common framework and connections to some signal processing algorithms are highlighted. These algorithms are grouped into two families; in the first one, the estimates converge to eigenvectors, and in the second one, a global convergence to a set of orthonormal bases of an eigenspace is achieved. In Section 3, after presenting a brief review of a general Gaussian approximation result, we shall focus exclusively on the first family in this paper, while a study of the second family will be the subject of a forthcoming paper. Closed-form expressions of the covariance in the limiting distributions of the eigenvector estimators in a constant step size environment are given by solving Lyapunov equations. Then, thanks to a continuity theorem, closed-form expressions of the covariance in the limiting distributions of the associated projection matrices are derived. These expressions are further analysed, compared with those obtained in batch estimation and some by-products as mean square errors are further derived. Finally we present in Section 4 some

simulation results with two purposes. First, we examine the accuracy of the expressions of the mean square error of eigenvectors and subspace projection matrix estimators, and investigate the domain of the step size for which the asymptotic approach is valid. Second, we examine performance criteria for which no general results are avalaible, such as the speed of convergence or the deviation from orthonormality. We evaluate the speed of convergence of the algorithms under study and lastly analyse several tradeoffs between the mean square error, the speed of convergence and the deviation from orthonormality.

The following notations are used throughout the paper. Matrices and vectors are represented by bold uppercase and boldlower case characters, respectively. Vectors are by default in column orientation. $\mathbf{e}_i^n$ is the ith unit vector in $\mathcal{R}^n$. T stands for transpose and $\mathbf{I}$ is the identity matrix. $\mathrm{E}(.), \mathrm{Cov}(.), \mathrm{Tr}(.)$ and $\|\,.\,\|_{\mathrm{Fro}}$ denote the expectation, the covariance, the trace operator and the Frobenius matrix norm, respectively. Vec(.) is the "vectorization" operator that turns a matrix into a vector consisting of the columns of the matrix stacked one below another. It is used in conjunction with the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ as the block matrix, the (i, j) block element of which is $a_{i,j}\mathbf{B}$. $\mathrm{Diag}(a_1, \ldots, a_n)$ is a diagonal matrix with diagonal elements a_i and $\mathrm{Diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ is a block diagonal matrix with block-diagonal matrices $\mathbf{A}_i$. The symbol 1_A denotes the indicator function of the condition A, that assumes the value 1 if this condition is satisfied and 0 otherwise.

2 Review of the algorithms under study

2.1 General structure

For a given $n \times n$ covariance matrix $\mathbf{R}_x = E(\mathbf{x}\mathbf{x}^T)$ of a Gaussian distributed, zero mean real random vector $\mathbf{x}$, denote by $\lambda_1 \geq \ldots \geq \lambda_n$ the eigenvalues of $\mathbf{R}_x$ and by $\mathbf{v}_1, \ldots, \mathbf{v}_n$ corresponding normalized eigenvectors. We tackle two kinds of problems. On the one hand, we are interested in adaptively estimating r normalized eigenvectors corresponding to the r largest [or smallest] distinct eigenvalues $(\lambda_1, \ldots, \lambda_r)$ [resp. $\lambda_{n-r+1}, \ldots, \lambda_n$] of $\mathbf{R}_x$. And on the other hand, we only consider the recursive updating of an (approximately) orthonormal basis of an r-dimensional dominant [or minorant] invariant subspace

of $\mathbf{R}_x$, where we only have to assume $\lambda_r > \lambda_{r+1}$ [resp. $\lambda_{n-r} > \lambda_{n-r+1}$]. Most of the stochastic algorithms introduced in the neural network community for estimating such eigenvectors or eigenspaces can be described in a common framework. They can be derived as a stochastic approximation algorithm, which can be seen as a counterpart of the "simultaneous iteration method" of numerical analysis [9]. This stochastic approximation algorithm reads

$$\mathbf{W}'_{t+1} = \mathbf{W}_t + \mathbf{R}_t \mathbf{W}_t \mathbf{\Gamma}_t \tag{2.1}$$

$$\mathbf{W}_{t+1} = \mathbf{W}'_{t+1} \mathbf{S}_{t+1}^{-1} \tag{2.2}$$

in which $\mathbf{W}_t = (\mathbf{w}_{t,1}, \ldots, \mathbf{w}_{t,r}) \in \mathcal{R}^{n \times r}$ is a matrix, the columns $\mathbf{w}_{t,k} \in \mathcal{R}^n$ of which are orthonormal and approximate r dominant eigenvectors of $\mathbf{R}_x$. In (2.1), the matrix $\mathbf{\Gamma}_t$ is the usual $r \times r$ diagonal gain matrix of stochastic approximation. We assume that $\mathbf{\Gamma}_t = \gamma_t \mathbf{I}_r$ except in one algorithm, where $\mathbf{\Gamma}_t = \gamma_t \mathrm{Diag}(1, \alpha_2, \ldots, \alpha_r)$ with $\alpha_i > 0$ is used in order to take into account a better tradeoff between the misadjustment and the speed of convergence. We suppose that the gain sequence γ_t satisfies the conditions: $\sum_{t=1}^{\infty} \gamma_t = +\infty$ and $\lim_{t \to +\infty} \gamma_t = 0$. The matrix $\mathbf{R}_t$ in (2.1) is an estimate of the covariance matrix $\mathbf{R}_x$. In all this paper, we shall use for $\mathbf{R}_t$ the instantaneous estimate $\mathbf{x}_t \mathbf{x}_t^T$.

In (2.2), $\mathbf{S}_{t+1}$ is a matrix depending on $\mathbf{W}'_{t+1}$, which orthonormalizes the columns of $\mathbf{W}'_{t+1}$. Thus, $\mathbf{W}_t$ has orthonormal columns for all t. Depending on the form of matrix $\mathbf{S}_{t+1}$, variants of the basic stochastic algorithm are obtained.

2.2 Dominant invariant subspace algorithms

Since the main problem addressed by the adaptive subspace algorithms introduced in the neural network literature is principal component analysis, these authors focused their attention on the dominant invariant subspace algorithms.

2.2.1 Algorithm converging to a rotated basis of an eigenvector subspace

The matrix $\mathbf{S}_{t+1}$ orthonormalizes the columns of $\mathbf{W}'_{t+1}$ in (2.2) in a symmetrical way. Since $\mathbf{W}_t$ has orthonormal columns, for small γ_t the columns of $\mathbf{W}'_{t+1}$ in (2.1) will be linearly independent, although not orthonormal. Then $\mathbf{W}'^T_{t+1}\mathbf{W}'_{t+1}$ is positive definite, and $\mathbf{W}_{t+1}$ will have orthonormal columns if $\mathbf{S}_{t+1} = (\mathbf{W}'^T_{t+1}\mathbf{W}'_{t+1})^{1/2}$. A stochastic algorithm denoted *Subspace Network Learning* (SNL) is obtained when, assuming γ_t is small, $\mathbf{S}^{-1}_{t+1}$ is expanded and when the term $O(\gamma_t^2)$ is omitted from its expansion. The algorithm reads

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \gamma_t[\mathbf{I}_n - \mathbf{W}_t\mathbf{W}_t^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{W}_t, \tag{2.3}$$

which can be written columnwise:

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - \sum_{i=1}^{r}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1, \ldots, r. \tag{2.4}$$

The convergence of this algorithm has been earlier studied in [10] and then in [11], where it is shown that the solution of its associated ODE needs not tend to the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$, but only to a rotated basis of the subspace spanned by them.

Written in the form $\mathbf{W}_{t+1} = \mathbf{W}_t + \gamma_t[\mathbf{x}_t\mathbf{x}_t^T - \mathbf{W}_t\mathbf{W}_t^T\mathbf{x}_t\mathbf{x}_t^T]\mathbf{W}_t$, the SNL algorithm is quite similar to the algorithm presented by Yang [12] and further analyzed in [13]. This latter algorithm is a stochastic gradient algorithm based on the unconstrained minimization of $\mathrm{E}\| \mathbf{x}_t - \mathbf{W}\mathbf{W}^T\mathbf{x}_t \|^2_{\mathrm{Fro}}$, and it reads:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \gamma_t[2\mathbf{x}_t\mathbf{x}_t^T - \mathbf{x}_t\mathbf{x}_t^T\mathbf{W}_t\mathbf{W}_t^T - \mathbf{W}_t\mathbf{W}_t^T\mathbf{x}_t\mathbf{x}_t^T]\mathbf{W}_t, \tag{2.5}$$

in which the term between brackets is the symmetrization of the term $\mathbf{x}_t\mathbf{x}_t^T - \mathbf{W}_t\mathbf{W}_t^T\mathbf{x}_t\mathbf{x}_t^T$ of the SNL algorithm. In [12], it is shown that like the SNL algorithm, the globally asymptotically stable solution of the associated ODE to (2.5) is the set of the orthonormal bases of the r-dominant invariant subspace of $\mathbf{R}_x$.

2.2.2 Algorithms converging to an eigenvector basis

Another starting point for deriving practical algorithms from (2.1) and (2.2) is that the matrix $\mathbf{S}_t$ performs the Gram-Schmidt orthogonalization on the columns of $\mathbf{W}'_t$. An algorithm, denoted *Stochastic Gradient Ascent* (SGA) algorithm, is obtained if the successive columns of matrix $\mathbf{W}_{t+1}$ are expanded, assuming γ_t small enough. By omitting the $O(\gamma_t^2)$ term in this expansion, we obtain

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - \mathbf{w}_{t,k}\mathbf{w}_{t,k}^T - 2\sum_{i=1}^{k-1}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1,\ldots,r. \tag{2.6}$$

An extension of this algorithm is obtained if $\mathbf{\Gamma}_t = \gamma_t \text{Diag}(\alpha_1, \alpha_2, \ldots, \alpha_r)$ with $\alpha_1 = 1$ and $\alpha_i > 0$,

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \alpha_k\gamma_t[\mathbf{I}_n - \mathbf{w}_{t,k}\mathbf{w}_{t,k}^T - \sum_{i=1}^{k-1}(1+\frac{\alpha_i}{\alpha_k})\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1,\ldots,r. \tag{2.7}$$

The so called *Generalized Hebbian Algorithm* (GHA) is derived from the SNL algorithm (2.3) by replacing the matrix $\mathbf{W}_t^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{W}_t$ of the SNL algorithm by its diagonal and superdiagonal only:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \gamma_t[\mathbf{x}_t\mathbf{x}_t^T\mathbf{W}_t - \mathbf{W}_t\text{upper}(\mathbf{W}_t^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{W}_t)] \tag{2.8}$$

in which the operator "upper" sets all subdiagonal elements of a matrix to zero. When written column-wise, this algorithm is similar to the SGA algorithm (2.6), with the difference that there is no coefficient 2 in the sum:

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - \sum_{i=1}^{k}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1,\ldots,r. \tag{2.9}$$

Oja *et al* [16] proposed an algorithm denoted *Weighted Subspace Algorithm* (WSA), which is similar to the SNL algorithm, except for the scalar parameters $\beta_1, \ldots, \beta_r$:

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - \sum_{i=1}^{r}\frac{\beta_k}{\beta_i}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1,\ldots,r, \tag{2.10}$$

with $0 < \beta_1 < \ldots < \beta_r$. If $\beta_i = 1$ for all i, this algorithm reduces to the SNL algorithm.

It was respectively established by Oja [14], Sanger [15] and Oja *et al* [17], that the only asymptotically stable points of the ODE associated respectively to the SGA, GHA and WSA algorithms are the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$. We note that the first vector ($k = 1$) estimated by the SGA and GHA algorithms,

and the vector ($r = k = 1$) estimated by the SNL and WSA algorithms gives the *Constrained Hebbian learning rule* of the basic PCA neuron introduced by Oja [18]

$$\mathbf{w}_{t+1,1} = \mathbf{w}_{t,1} + \gamma_t[\mathbf{I}_n - \mathbf{w}_{t,1}\mathbf{w}_{t,1}^T]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,1}. \tag{2.11}$$

This algorithm also coincides with the algorithm denoted *Direct Adaptive Subspace Estimator*, which was proposed by Riou *et al* [19] for $k = 1$. This latter algorithm reads

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - (\mathbf{I}_n - \sum_{i=1}^{k-1}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T)(\mathbf{w}_{t,k}\mathbf{w}_{t,k}^T)]\mathbf{x}_t\mathbf{x}_t^T\mathbf{w}_{t,k} \quad \text{for } k = 1,\ldots,r, \tag{2.12}$$

and converges, after normalization of $\mathbf{w}_{t,k}$ for all k, $k > 1$, to the eigenvectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$.

2.3 Minorant invariant subspace algorithm

Minor component analysis was also considered in neural network to solve the problem of optimal fitting in the total least square sense. Xu *et al.* [20] introduced the *Optimal Fitting Analyzer* (OFA) algorithm by modifying the SGA algorithm. This algorithm reads

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t[\mathbf{I}_n - \mathbf{x}_t\mathbf{x}_t^T + \mathbf{w}_{t,k}\mathbf{w}_{t,k}^T\mathbf{x}_t\mathbf{x}_t^T - \mathbf{w}_{t,k}\mathbf{w}_{t,k}^T - \beta\sum_{i=k+1}^{n}\mathbf{w}_{t,i}\mathbf{w}_{t,i}^T\mathbf{x}_t\mathbf{x}_t^T]\mathbf{w}_{t,k} \quad \text{for } k = n-r+1,\ldots,n. \tag{2.13}$$

Oja [2] showed that, under the conditions that the eigenvalues are distinct, and that

$$\lambda_{n-r+1} < 1 \quad \text{and} \quad \beta > \frac{\lambda_{n-r+1}}{\lambda_n} - 1, \tag{2.14}$$

the only asymptotically stable points of the associated ODE are the eigenvectors $\mathbf{v}_{n-r+1},\ldots,\mathbf{v}_n$. Note that the magnitude of the eigenvalues must be controlled in practice by normalizing $\mathbf{x}_t$ so that the expression between brackets in (2.13) becomes homogeneous.

3 Asymptotic performance analysis

3.1 A short review of a general Gaussian approximation result

In this section, we evaluate the asymptotic distributions of eigenvector and subspace projection matrix estimators given by the previous algorithms. For this purpose, we shall use the following result [21, Th.

2, p. 108]. Consider a constant step size recursive stochastic algorithm

$$\Theta_{t+1} = \Theta_t + \gamma f(\Theta_t, \mathbf{x}_t) \tag{3.15}$$

with $\mathbf{x}_t = g(\xi_t)$, where ξ_t is a Markov chain independent of Θ_t. Suppose that the parameter vector Θ_t converges almost surely to the unique asymptotically stable point Θ_* in the corresponding decreasing step size algorithm. Consider the continuous Lyapunov equation

$$\mathbf{D}\mathbf{C}_\Theta + \mathbf{C}_\Theta\mathbf{D}^T + \mathbf{G} = \mathbf{O} \tag{3.16}$$

where $\mathbf{D}$ and $\mathbf{G}$ are, respectively, the derivative of the mean field and the covariance of the field of the algorithm (3.15)

$$\mathbf{D} \stackrel{\text{def}}{=} \mathrm{E}[\frac{\partial f}{\partial \Theta}(\Theta, \mathbf{x}_t)]_{\Theta=\Theta_*}, \tag{3.17}$$

$$\mathbf{G} \stackrel{\text{def}}{=} \sum_{t=-\infty}^{\infty} \mathrm{Cov}[f(\Theta_*, \mathbf{x}_t), f(\Theta_*, \mathbf{x}_0)]. \tag{3.18}$$

If all the eigenvalues of the derivative $\mathbf{D}$ of the mean field have strictly negative real parts, then, in a stationary situation, when $\gamma \to 0$ and $t \to \infty$, we have the convergence in distribution

$$\frac{1}{\sqrt{\gamma}}(\Theta_t - \Theta_*) \stackrel{\mathcal{L}}{\to} \mathcal{N}(0, \mathbf{C}_\Theta), \tag{3.19}$$

where $\mathbf{C}_\Theta$ is the unique symmetric solution of the Lyapunov equation (3.16).

3.2 Asymptotic distributions of eigenvector estimators

To characterize the derivative of the mean field and the covariance of the field of the SGA, GHA, WSA and OFA algorithms, we use the Vec operator which turns the $n \times r$ matrix $\mathbf{W}$ into the $nr \times 1$ vector parameter Vec$\mathbf{W}$. Thus the four algorithms (2.7), (2.9), (2.10) and (2.13), which read

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma f_k(\mathbf{W}_t, \mathbf{x}_t\mathbf{x}_t^T) \tag{3.20}$$

for $k = 1, \ldots, r$ [resp. $k = n - r + 1, \ldots, n$] for the SGA, GHA, WSA [resp. OFA] algorithms, can be written in a form similar to that of the equation (3.15):

$$\mathrm{Vec}\mathbf{W}_{t+1} = \mathrm{Vec}\mathbf{W}_t + \gamma f(\mathrm{Vec}\mathbf{W}_t, \mathbf{x}_t\mathbf{x}_t^T). \tag{3.21}$$

3.2.1 Local characterization of the field

Derivative of the field Let us denote by $\mathbf{D}_{i,j}$ the (i,j) block $\mathrm{E}[\frac{\partial f_i(\mathbf{W},\mathbf{x}_t\mathbf{x}_t^T)}{\partial \mathbf{w}_j}]_{\mathbf{W}=\mathbf{W}_*}$ of the $nr \times nr$ block matrix $\mathbf{D}$, for $(i,j) \in \{1,\ldots,r\}^2$ [resp. $(i,j) \in \{n-r+1,\ldots,n\}^2$] of the SGA, GHA and WSA [resp., OFA] algorithms. Since the field f_k of definition (3.20) is linear in its second argument, the mean field at any point $\mathbf{W}$ is simply

$$\mathrm{E}(f_i(\mathbf{W},\mathbf{x}_t\mathbf{x}_t^T)) = f_i(\mathbf{W},\mathrm{E}(\mathbf{x}_t\mathbf{x}_t^T)) = f_i(\mathbf{W},\mathbf{R}_x) \tag{3.22}$$

If we note that

$$\frac{\partial \mathbf{R}_x\mathbf{w}_j}{\partial \mathbf{w}_j} = \mathbf{R}_x, \qquad \text{and} \qquad \frac{\partial \mathbf{w}_i\mathbf{w}_i^T\mathbf{R}_x\mathbf{w}_j}{\partial \mathbf{w}_j} = \mathbf{w}_i\mathbf{w}_i^T\mathbf{R}_x \qquad \text{for } i \neq j$$

$$\frac{\partial \mathbf{w}_j\mathbf{w}_j^T\mathbf{R}_x\mathbf{w}_i}{\partial \mathbf{w}_j} = \begin{cases} (\mathbf{w}_j^T\mathbf{R}_x\mathbf{w}_i)\mathbf{I}_n + \mathbf{w}_j\mathbf{w}_i^T\mathbf{R}_x & \text{for } i \neq j \\ (\mathbf{w}_i^T\mathbf{R}_x\mathbf{w}_i)\mathbf{I}_n + 2\mathbf{w}_i\mathbf{w}_i^T\mathbf{R}_x & \text{for } i = j \end{cases}$$

whose values at $\mathbf{W} = \mathbf{W}_* = (\mathbf{v}_1,\ldots,\mathbf{v}_r)$, are respectively, $\mathbf{R}_x$, $\lambda_i\mathbf{v}_i\mathbf{v}_i^T$, $\lambda_i\mathbf{v}_j\mathbf{v}_i^T$ and $\lambda_i\mathbf{I}_n + 2\lambda_i\mathbf{v}_i\mathbf{v}_i^T$, it is easy to obtain the following results for the SGA, GHA, WSA and OFA algorithms, respectively

$$\mathbf{D}_{i,j}^{SGA} = \begin{cases} -\alpha_i[\sum_{k=1}^{i-1}(\lambda_i + \frac{\alpha_k}{\alpha_i}\lambda_k)\mathbf{v}_k\mathbf{v}_k^T + 2\lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{k=i+1}^{n}(\lambda_i - \lambda_k)\mathbf{v}_k\mathbf{v}_k^T] & i = j \\ \mathbf{O} & i < j \\ -\alpha_i(1 + \frac{\alpha_j}{\alpha_i})\lambda_i\mathbf{v}_j\mathbf{v}_i^T & i > j \end{cases} \tag{3.23}$$

$$\mathbf{D}_{i,j}^{GHA} = \begin{cases} -[\sum_{k=1}^{i-1}\lambda_i\mathbf{v}_k\mathbf{v}_k^T + 2\lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{k=i+1}^{n}(\lambda_i - \lambda_k)\mathbf{v}_k\mathbf{v}_k^T] & i = j \\ \mathbf{O} & i < j \\ -\lambda_i\mathbf{v}_j\mathbf{v}_i^T & i > j \end{cases} \tag{3.24}$$

$$\mathbf{D}_{i,j}^{WSA} = \begin{cases} -[\sum_{\substack{k=1 \\ k\neq i}}^{r}(\lambda_i - (1 - \frac{\beta_i}{\beta_k})\lambda_k)\mathbf{v}_k\mathbf{v}_k^T + 2\lambda_i\mathbf{v}_i\mathbf{v}_i^T + \sum_{k=r+1}^{n}(\lambda_i - \lambda_k)\mathbf{v}_k\mathbf{v}_k^T] & i = j \\ -\frac{\beta_i}{\beta_j}\lambda_i\mathbf{v}_j\mathbf{v}_i^T & i \neq j \end{cases} \tag{3.25}$$

$$\mathbf{D}_{i,j}^{OFA} = \begin{cases} \sum_{k=1}^{i-1}(\lambda_i - \lambda_k)\mathbf{v}_k\mathbf{v}_k^T + 2(\lambda_i - 1)\mathbf{v}_i\mathbf{v}_i^T + \sum_{k=i+1}^{n}(\lambda_i - (1+\beta)\lambda_k)\mathbf{v}_k\mathbf{v}_k^T & i = j \\ -\beta\lambda_i\mathbf{v}_j\mathbf{v}_i^T & i < j \\ \mathbf{O} & i > j \end{cases} \tag{3.26}$$

From these expressions, the following theorem is proved in the Appendix.

Theorem 1 *The eigenvalues of the derivative* $\mathbf{D}$ *of the mean field of the SGA, GHA and OFA algorithms are strictly negative real, and those of the WSA algorithm have strictly negative real parts.*

Covariance of the field In case the observations $\mathbf{x}_t$ are independent, the covariance of the field (3.18) evaluated in $\mathbf{W} = \mathbf{W}_*$ is an $nr \times nr$ block matrix $\mathbf{G}$, the (i, j) block element of which is

$$\mathbf{G}_{i,j} = \mathrm{E}[f_i(\mathbf{W}_*, \mathbf{x}_t\mathbf{x}_t^T) f_j^T(\mathbf{W}_*, \mathbf{x}_t\mathbf{x}_t^T)]. \tag{3.27}$$

We note that the field $f_i(\mathbf{W}_*, \mathbf{x}_t\mathbf{x}_t^T)$ reduces to a linear expression in $\mathbf{x}_t\mathbf{x}_t^T$: $\mathbf{B}_i\mathbf{x}_t\mathbf{x}_t^T\mathbf{v}_i$ for the SGA, GHA and WSA algorithms, and to an affine expression in $\mathbf{x}_t\mathbf{x}_t^T$: $\mathbf{a}_i + \mathbf{B}_i\mathbf{x}_t\mathbf{x}_t^T\mathbf{v}_i$ for the OFA algorithm, with $\mathbf{B}_i$ depending on the algorithm. Thanks to the classic property:

$$\mathrm{Vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\mathrm{Vec}(\mathbf{B}), \tag{3.28}$$

$\mathbf{B}_i\mathbf{x}_t\mathbf{x}_t^T\mathbf{v}_i = (\mathbf{v}_i^T \otimes \mathbf{B}_i)\mathrm{Vec}(\mathbf{x}_t\mathbf{x}_t^T)$. Therefore, $\mathbf{G}_{i,j}$ reads

$$\mathbf{G}_{i,j} = (\mathbf{v}_i^T \otimes \mathbf{B}_i)\mathrm{Cov}(\mathrm{Vec}(\mathbf{x}_t\mathbf{x}_t^T))(\mathbf{v}_j \otimes \mathbf{B}_j^T), \tag{3.29}$$

since moreover; $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$. Now, for a Gaussian vector $\mathbf{x}$, we have [25, p. 57]

$$\mathrm{Cov}(\mathrm{Vec}(\mathbf{x}\mathbf{x}^T)) = \mathbf{R}_x \otimes \mathbf{R}_x + (\mathbf{R}_x \otimes \mathbf{R}_x)\mathbf{K}, \tag{3.30}$$

where $\mathbf{K}$ is an $n^2 \times n^2$ block matrix, acting as a permutation operator, in the sense that for any vector $\mathbf{a}$ or matrix $\mathbf{A}$ and vector $\mathbf{b}$, we have

$$\mathbf{K}(\mathbf{a} \otimes \mathbf{b}) = \mathbf{b} \otimes \mathbf{a} \qquad \text{and} \qquad \mathbf{K}(\mathbf{A} \otimes \mathbf{b}) = \mathbf{b} \otimes \mathbf{A}. \tag{3.31}$$

It follows that

$$\mathbf{G}_{i,j} = (\mathbf{v}_i^T \otimes \mathbf{B}_i)(\mathbf{R}_x \otimes \mathbf{R}_x + (\mathbf{R}_x \otimes \mathbf{R}_x)\mathbf{K})(\mathbf{v}_j \otimes \mathbf{B}_j^T), \tag{3.32}$$

thanks to (3.31) and to the classic property

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC} \otimes \mathbf{BD}). \tag{3.33}$$

Therefore, $\mathbf{G}_{i,j}$ becomes

$$\begin{aligned}\mathbf{G}_{i,j} &= \mathbf{v}_i^T\mathbf{R}_x\mathbf{v}_j \otimes \mathbf{B}_i\mathbf{R}_x\mathbf{B}_j^T + \mathbf{v}_i^T\mathbf{R}_x\mathbf{B}_j^T \otimes \mathbf{B}_i\mathbf{R}_x\mathbf{v}_j \\ &= \lambda_i 1_{i=j}\mathbf{B}_i\mathbf{R}_x\mathbf{B}_j^T + \lambda_i\lambda_j\mathbf{v}_i^T\mathbf{B}_j^T \otimes \mathbf{B}_i\mathbf{v}_j \end{aligned} \tag{3.34}$$

Last, taking into account the different values of $\mathbf{B}_i$, and thanks to the relation $\mathbf{v}_i^T \otimes \mathbf{v}_j = \mathbf{v}_j\mathbf{v}_i^T$, it is straightforward to obtain the values of the blocks $\mathbf{G}_{i,j}$ for the SGA, GHA, WSA and OFA algorithms respectively

$$\mathbf{G}_{i,j}^{SGA} = \begin{cases} \sum_{k=1}^{i-1}\alpha_k^2\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T + \sum_{k=i+1}^{n}\alpha_i^2\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T] & i = j \\ -\alpha_i^2\lambda_i\lambda_j\mathbf{v}_j\mathbf{v}_i^T & i < j \\ -\alpha_j^2\lambda_j\lambda_i\mathbf{v}_j\mathbf{v}_i^T & i > j \end{cases} \tag{3.35}$$

$$\mathbf{G}_{i,j}^{GHA} = \begin{cases} \sum_{k=i+1}^{n}\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T & i = j \\ \mathbf{O} & i \neq j \end{cases} \tag{3.36}$$

$$\mathbf{G}_{i,j}^{WSA} = \begin{cases} \sum_{k=1}^{r}\lambda_i\lambda_k(1-\frac{\beta_i}{\beta_k})^2\mathbf{v}_k\mathbf{v}_k^T + \sum_{k=r+1}^{n}\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T & i = j \\ (1-\frac{\beta_i}{\beta_j})(1-\frac{\beta_j}{\beta_i})\lambda_i\lambda_j\mathbf{v}_j\mathbf{v}_i^T & i \neq j \end{cases} \tag{3.37}$$

$$\mathbf{G}_{i,j}^{OFA} = \begin{cases} \sum_{k=1}^{i-1}\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T + (1+\beta)^2\sum_{k=i+1}^{n}\lambda_i\lambda_k\mathbf{v}_k\mathbf{v}_k^T & i = j \\ (1+\beta)\lambda_i\lambda_j\mathbf{v}_j\mathbf{v}_i^T & i \neq j \end{cases} \tag{3.38}$$

3.2.2 Solution of the Lyapunov equation

For independent observations $\mathbf{x}_t$ and for the investigated algorithms, which can be written in a form similar to (3.15) with $\xi_t = \mathbf{x}_t$ for which the derivative of the mean field have strictly negative real parts (Theorem 1), the hypotheses of the model of Benveniste *et al* ([21, Th. 2, p. 108]) are fulfilled. However, the underlying assumption for the results by Benveniste *et al* is that the solution of the corresponding stochastic approximation type algorithms with decreasing step size, almost surely converges to the unique asymptotically stable solution of the associated ODE. Since the normalized eigenvectors are defined up to a sign, the global attractor $\mathbf{W}_*$ is not unique. However, the practical use of the Benveniste results in such situation is usually justified (for example in [22]) by using formally a general approximation result

([21, Th. 1, p. 107]). Furthermore, the almost sure convergence of the associated decreasing step size algorithms are not strictly fulfilled for the SGA, GHA, WSA and OFA algorithms. This a.s. convergence would need a boundedness condition, whose satisfaction is a challenging problem. However, as it is discussed in [23], this condition was proved for only the algorithm (2.11), where Oja *et al* [24] showed that if this algorithm is used with uniformly bounded inputs $\mathbf{x}_t$, then $\mathbf{w}_{t,1}$ remains inside some bounded subset. If we allow ourselves the Benveniste results in our situation, the Lyapunov continuous equations can be solved exactly. The following theorem is proved in the Appendix.

Theorem 2 *The covariance matrices* $\mathbf{C}_W$ *of the asymptotic distribution that appears in (3.19) read*

$$\mathbf{C}_W = \sum_{\substack{1\leq i\leq r\\ 1\leq k\neq i\leq n}} b_{k,i}(\mathbf{e}_i^r\mathbf{e}_i^{r^T}\otimes\mathbf{v}_k\mathbf{v}_k^T) + \sum_{1\leq i\neq j\leq r} c_{i,j}(\mathbf{e}_i^r\mathbf{e}_j^{r^T}\otimes\mathbf{v}_j\mathbf{v}_i^T) \tag{3.39}$$

with for the SGA, GHA, WSA and OFA algorithms respectively

$$b_{k,i}^{SGA} = \frac{\alpha_k\lambda_i\lambda_k}{2(\lambda_k-\lambda_i)}1_{k<i} + \frac{\alpha_i\lambda_i\lambda_k}{2(\lambda_i-\lambda_k)}1_{k>i} \qquad c_{i,j}^{SGA} = -\frac{\alpha_i\lambda_j\lambda_i}{2(\lambda_i-\lambda_j)}1_{i<j} - \frac{\alpha_j\lambda_i\lambda_j}{2(\lambda_j-\lambda_i)}1_{i>j} \tag{3.40}$$

$$b_{k,i}^{GHA} = \frac{\lambda_i^2}{2(\lambda_k-\lambda_i)}1_{k<i} + \frac{\lambda_i\lambda_k}{2(\lambda_i-\lambda_k)}1_{k>i} \qquad c_{i,j}^{GHA} = -\frac{\lambda_j^2}{2(\lambda_i-\lambda_j)}1_{i<j} - \frac{\lambda_i^2}{2(\lambda_j-\lambda_i)}1_{i>j} \tag{3.41}$$

$$b_{k,i}^{WSA} = \lambda_i\lambda_k c^{ki}1_{k<i} + \lambda_i\lambda_k b^{ik}1_{i<k\leq r} + \frac{\lambda_i\lambda_k}{2(\lambda_i-\lambda_k)}1_{k>r} \quad c_{i,j}^{WSA} = \lambda_i\lambda_j d^{ij}1_{i<j} + \lambda_i\lambda_j d^{ji}1_{i>j} \tag{3.42}$$

$$b_{k,i}^{OFA} = \frac{\lambda_i\lambda_k}{2(\lambda_k-\lambda_i)}1_{k<i} + g^{ik}1_{k>i} \qquad c_{i,j}^{OFA} = h^{ij}1_{i<j} + h^{ji}1_{i>j}, \tag{3.43}$$

and where b^{ij}, c^{ij}, d^{ij}, g^{ij} *and* h^{ij} *are defined in the proof. For the OFA algorithm, the two summations are respectively over* $n-r+1\leq i\leq n, 1\leq k\neq i\leq n$ *and* $n-r+1\leq i\neq j\leq n$.

Remark Of course, if $k=r=1$, the covariance matrices $\mathbf{C}_W$ of the SGA, GHA and WSA algorithms coincide with the expression of the covariance matrix of the Oja rule as given by Yang [6, Eq. (26)]:

$$\mathbf{C}_W = \sum_{k=2}^{n} \frac{\lambda_1\lambda_k}{2(\lambda_1-\lambda_k)}\mathbf{v}_k\mathbf{v}_k^T. \tag{3.44}$$

For the WSA algorithm, we note that when β_i tends to 1 for all i, b^{ij}, c^{ij} and d^{ij} tend to 0 from (B.30). Therefore, $\mathbf{C}_W$ tends to the finite value:

$$\mathbf{C}_W = \sum_{1\leq i\leq r<k\leq n} \frac{\lambda_i\lambda_k}{2(\lambda_i-\lambda_k)}(\mathbf{e}_i^r\mathbf{e}_i^{r^T}\otimes\mathbf{v}_k\mathbf{v}_k^T), \tag{3.45}$$

whereas when $\beta_i = 1$, for $i = 1, \ldots, r$ the WSA algorithm coincides with the SNL algorithm, which does not converge to the eigenvectors. We will show in Section 4, that if $\mathbf{C}_W$ keeps a finite value, the speed of convergence worsens when all the parameters β_i tend to 1. Since in many applications we are interested in the associated projection matrix estimators $\mathbf{P}_t = \mathbf{W}_t \mathbf{W}_t^T$, we consider now its asymptotic distribution.

3.2.3 Asymptotic distributions of projection matrix estimators

The tool we use is a continuity theorem that can be directly adapted from the classic theorem (see [26, Th. 6.2a p. 387]). Applying this theorem to the differentiable mapping $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_r) \to \mathbf{P} = \sum_{k=1}^r \mathbf{w}_k \mathbf{w}_k^T$ gives the asymptotic distribution of subspace projector matrix estimator $\mathbf{P}_t$ for the different algorithms

$$\frac{1}{\sqrt{\gamma}}(\mathrm{Vec}(\mathbf{P}_t) - \mathrm{Vec}(\mathbf{P}_*)) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{O}, \mathbf{C}_P), \tag{3.46}$$

when $\gamma \to 0$ and $t \to \infty$. In (3.46), $\mathbf{P}_* \stackrel{\text{def}}{=} \sum_{k=1}^r \mathbf{v}_k \mathbf{v}_k^T$, and $\mathbf{C}_P$ is equal to

$$\mathbf{C}_P = \frac{d\mathrm{Vec}(\mathbf{P})}{d\mathrm{Vec}(\mathbf{W})} \mathbf{C}_W \frac{d^T \mathrm{Vec}(\mathbf{P})}{d\mathrm{Vec}(\mathbf{W})}_{\mathbf{W}=\mathbf{W}_*} \tag{3.47}$$

with

$$\frac{d\mathrm{Vec}(\mathbf{P})}{d\mathrm{Vec}(\mathbf{W})} = (\mathbf{I}_n \otimes \mathbf{w}_1 + \mathbf{w}_1 \otimes \mathbf{I}_n, \cdots, \mathbf{I}_n \otimes \mathbf{w}_r + \mathbf{w}_r \otimes \mathbf{I}_n). \tag{3.48}$$

Of course, the previous results also apply to the OFA algorithm with $\mathbf{W} = (\mathbf{w}_{n-r+1}, \ldots, \mathbf{w}_n)$. Therefore

$$\mathbf{C}_P = \sum_{1 \le i,j \le r} (\mathbf{I}_n \otimes \mathbf{v}_i + \mathbf{v}_i \otimes \mathbf{I}_n) \mathbf{C}_{W_{i,j}} (\mathbf{I}_n \otimes \mathbf{v}_j^T + \mathbf{v}_j^T \otimes \mathbf{I}_n) \tag{3.49}$$

where $\mathbf{C}_{W_{i,j}}$ denotes the (i,j) block of the $nr \times nr$ block matrix $\mathbf{C}_W$. From Theorem 2

$$\mathbf{C}_{W_{i,i}} = \sum_{1 \le k \ne i \le n} b_{k,i} \mathbf{v}_k \mathbf{v}_k^T \quad \text{and} \quad \mathbf{C}_{W_{i,j}} = c_{i,j} \mathbf{v}_j \mathbf{v}_i^T. \tag{3.50}$$

Thanks to (3.33) and the relation $\mathbf{v}_j \mathbf{v}_i^T = \mathbf{v}_j \otimes \mathbf{v}_i^T = \mathbf{v}_i^T \otimes \mathbf{v}_j$, in case of the SGA, GHA or WSA algorithms, (3.49) reads

$$\mathbf{C}_P = \sum_{1 \le i \le j \le n} (b_{j,i} 1_{i \le r < j} + d_{i,j} 1_{i < j \le r})(\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)(\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)^T \tag{3.51}$$

with $d_{i,j} \stackrel{\text{def}}{=} c_{i,j} + c_{j,i} + b_{i,j} + b_{j,i}$. Of course, (3.51) also holds in case of the OFA algorithm, where the two indicators are over $j < n-r+1 \leq i$ and $n-r+1 \leq i < j$ respectively. From the expressions of $b_{i,j}$ and $c_{i,j}$ derived from (3.40), (3.41), (3.42) and (3.43), the terms $d_{i,j}$ read for the SGA, GHA, WSA and OFA algorithms, respectively:

$$b_{j,i}^{SGA} = \frac{\alpha_i \lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} \qquad d_{i,j}^{SGA} = 0 \tag{3.52}$$

$$b_{j,i}^{GHA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} \qquad d_{i,j}^{GHA} = \frac{\lambda_j}{2} \tag{3.53}$$

$$b_{j,i}^{WSA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} \qquad d_{i,j}^{WSA} = \lambda_i \lambda_j (b^{ij} + c^{ij} + 2d^{ij}) \tag{3.54}$$

$$b_{j,i}^{OFA} = \frac{\lambda_i \lambda_j}{2(\lambda_j - \lambda_i)} \qquad d_{i,j}^{OFA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} + 2h^{ij} + g^{ij} \tag{3.55}$$

where b^{ij}, c^{ij}, d^{ij} , g^{ij} and h^{ij} are defined in the Proof of Theorem 2. For the WSA algorithm, we note that when β_i tends to 1 for all i, $d_{i,j}$ tends to 0, because of (B.30). Therefore, $\mathbf{C}_P$ which is given by (3.51) tends to

$$\mathbf{C}_P = \sum_{1 \leq i \leq r < j \leq n} \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)(\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)^T, \tag{3.56}$$

which is an expression that coincides with the covariance in distribution of the projection matrix estimator of the Yang algorithm [8]. This property will be explained in a forthcoming paper.

3.3 Analysis of the results

First, the expressions (3.39), (3.51) and (3.56) can be compared with the covariances in the asymptotic distributions obtained in batch estimation. We know from ([25, Th. 13.5.1 p. 541]) that if $\mathbf{W}_t = (\mathbf{w}_{t,1}, \ldots, \mathbf{w}_{t,r})$ denotes the eigenvector matrix computed from the eigenvalue decomposition of the sample covariance matrix $\frac{1}{t}\sum_{k=1}^{t} \mathbf{x}_k \mathbf{x}_k^T$, then

$$\sqrt{t}\ (\text{Vec}(\mathbf{W}_t) - \text{Vec}(\mathbf{W}_*)) \stackrel{\mathcal{L}}{\rightarrow} \mathcal{N}(\mathbf{O}, \mathbf{C}_W) \tag{3.57}$$

when $t \rightarrow \infty$, provided $\lambda_1, \ldots, \lambda_{r+1}$ are distinct. In (3.57), $\mathbf{C}_W$ is equal to

$$\mathbf{C}_W = \sum_{\substack{1 \leq i \leq r \\ 1 \leq k \neq i \leq n}} \frac{\lambda_i \lambda_k}{(\lambda_i - \lambda_k)^2} (\mathbf{e}_i \mathbf{e}_i^T \otimes \mathbf{v}_k \mathbf{v}_k^T) - \sum_{1 \leq i \neq j \leq r} \frac{\lambda_i \lambda_j}{(\lambda_i - \lambda_j)^2} (\mathbf{e}_i \mathbf{e}_j^T \otimes \mathbf{v}_j \mathbf{v}_i^T) \tag{3.58}$$

in close similarity to (3.39). Similarly, if $\mathbf{P}_t = \sum_{1\leq i\leq r} \mathbf{w}_{t,i}\mathbf{w}_{t,i}^T$ denotes the batch estimated orthogonal projection matrix, we have, from [8]

$$\sqrt{t}\ (\mathrm{Vec}(\mathbf{P}_t) - \mathrm{Vec}(\mathbf{P}_*)) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{O}, \mathbf{C}_P) \tag{3.59}$$

with

$$\mathbf{C}_P = \sum_{1\leq i\leq r<j\leq n} \frac{\lambda_i\lambda_j}{(\lambda_i-\lambda_j)^2}(\mathbf{v}_i\otimes\mathbf{v}_j + \mathbf{v}_j\otimes\mathbf{v}_i)(\mathbf{v}_i\otimes\mathbf{v}_j + \mathbf{v}_j\otimes\mathbf{v}_i)^T \tag{3.60}$$

which is also in close similarity to (3.56) and to the first term of the summation (3.51). We note that unlike the expression (3.39) for $\mathbf{C}_W$, (3.51) for $\mathbf{C}_P$ is in fact an eigenvalue decomposition. This property will be used further in the paper.

Second, a simple global measure of performance of our adaptive algorithms is the MSE between $\mathbf{W}_t$ and $\mathbf{W}_*$, and between $\mathbf{P}_t$ and $\mathbf{P}_*$. These MSE can be obtained from the asymptotic distribution of $\mathrm{Vec}(\mathbf{W}_t)$ and of $\mathrm{Vec}(\mathbf{P}_t)$, if we suppose that both the first and second moments of the limiting distribution of $\frac{1}{\sqrt{\gamma}}(\mathbf{W}_t - \mathbf{W}_*)$ are equal to the corresponding asymptotic moments. In batch estimation, both the first and second moments are identical ([27, Theorem 9.24 p. 343]). Motivated by this observation, we postulate that this property also holds in our adaptive estimation. Therefore, $\|\,\mathrm{E}(\mathbf{W}_t) - \mathbf{W}_*\,\|^2_{\mathrm{Fro}} = o(\gamma)$ and $\mathrm{Cov}(\mathrm{Vec}\mathbf{W}_t) = \gamma\mathbf{C}_W + o(\gamma)$, and by expanding $\mathbf{P}$ around $\mathbf{P}_*$, $\|\,\mathrm{E}(\mathbf{P}_t) - \mathbf{P}_*\,\|^2_{\mathrm{Fro}} = o(\gamma)$, and $\mathrm{Cov}(\mathrm{Vec}\mathbf{P}_t) = \gamma\mathbf{C}_P + o(\gamma)$. Thus the MSE between $\mathbf{W}_t$ and $\mathbf{W}_*$ and between $\mathbf{P}_t$ and $\mathbf{P}_*$ is given respectively, by the trace of the covariance matrix in the asymptotic distribution of $\mathbf{W}_t$ and of $\mathbf{P}_t$

$$\mathrm{E}\|\,\mathbf{W}_t - \mathbf{W}_*\,\|^2_{\mathrm{Fro}} = \gamma\mathrm{Tr}(\mathbf{C}_W) + o(\gamma) \qquad \text{and} \qquad \mathrm{E}\|\,\mathbf{P}_t - \mathbf{P}_*\,\|^2_{\mathrm{Fro}} = \gamma\mathrm{Tr}(\mathbf{C}_P) + o(\gamma). \tag{3.61}$$

Since the trace is invariant under the orthonormal change of basis (B.1), $\mathrm{Tr}(\mathbf{C}_W) = \mathrm{Tr}(\mathbf{\Sigma}')$. From the expressions of $\mathbf{\Sigma}_1'^i$ and $\mathbf{\Sigma}_2'^{i,j}$ given in the proof of Theorem 2, the trace of $\mathbf{C}_W$ is equal respectively for the SGA, GHA, WSA and OFA algorithms to

$$\mathrm{Tr}(\mathbf{C}_W) = \sum_{i=1}^{r}\sum_{\substack{j=1\\ j\neq i}}^{n} \alpha_{min(i,j)}\frac{\lambda_i\lambda_j}{2|\lambda_j-\lambda_i|} \tag{3.62}$$

$$= \sum_{i=1}^{r}\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{\lambda_{max(i,j)}\lambda_i}{|\lambda_j-\lambda_i|} \tag{3.63}$$

$$= \sum_{1\le i\le r<j\le n} \frac{\lambda_i\lambda_j}{2(\lambda_i-\lambda_j)} + \sum_{1\le i<j\le r} \lambda_i\lambda_j(b^{ij}+c^{ij}) \tag{3.64}$$

$$= \frac{1}{2}\sum_{i=n-r+1}^{n}\sum_{j=1}^{i-1}\frac{\lambda_i\lambda_j}{\lambda_j-\lambda_i} + \sum_{n-r+1\le i<j\le n} g^{ij}. \tag{3.65}$$

As for $\mathrm{Tr}(\mathbf{C}_P)$, using (3.51) and the relation $\mathrm{Tr}(\mathbf{v}_i\mathbf{v}_j^T \otimes \mathbf{v}_k\mathbf{v}_l^T) = 1_{i=j}1_{k=l}$, we get respectively, for the SGA, GHA, WSA and OFA algorithms

$$\mathrm{Tr}(\mathbf{C}_P) = \sum_{1\le i\le r<j\le n} \alpha_i\frac{\lambda_i\lambda_j}{\lambda_i-\lambda_j} + 0 \tag{3.66}$$

$$= \sum_{1\le i\le r<j\le n} \frac{\lambda_i\lambda_j}{\lambda_i-\lambda_j} + \sum_{i=1}^{r} i\lambda_i \tag{3.67}$$

$$= \sum_{1\le i\le r<j\le n} \frac{\lambda_i\lambda_j}{\lambda_i-\lambda_j} + 2\sum_{1\le i<j\le r} \lambda_i\lambda_j(b^{ij}+c^{ij}+2d^{ij}) \tag{3.68}$$

$$= \sum_{1\le j<n-r+1\le i\le n} \frac{\lambda_i\lambda_j}{\lambda_j-\lambda_i} + \sum_{n-r+1\le i<j\le n} \frac{\lambda_i\lambda_j}{\lambda_j-\lambda_i} + 4h^{ij} + 2g^{ij}. \tag{3.69}$$

Finally, a finer picture of the MSE of $\mathbf{C}_P$ can be derived from the regular structure (3.51) of the covariance matrix $\mathbf{C}_P$ by decomposing the error $\mathbf{P}_t - \mathbf{P}_*$ into three terms

$$\mathbf{P}_t - \mathbf{P}_* = \mathbf{P}_{1,t} + \mathbf{P}_{2,t} + \mathbf{P}_{3,t}, \tag{3.70}$$

with

$$\mathbf{P}_{1,t} \stackrel{\text{def}}{=} \mathbf{P}_*(\mathbf{P}_t-\mathbf{P}_*)\mathbf{P}_*, \quad \mathbf{P}_{2,t} \stackrel{\text{def}}{=} \mathbf{P}_*\mathbf{P}_t\mathbf{P}_*^\perp + \mathbf{P}_*^\perp\mathbf{P}_t\mathbf{P}_*, \quad \text{and} \quad \mathbf{P}_{3,t} \stackrel{\text{def}}{=} \mathbf{P}_*^\perp\mathbf{P}_t\mathbf{P}_*^\perp. \tag{3.71}$$

Using $\mathbf{I}_n = \mathbf{P}_* + \mathbf{P}_*^\perp$, this is easily seen to be an orthogonal decomposition:

$$\| \mathbf{P}_t - \mathbf{P}_* \|_{\text{Fro}}^2 = \| \mathbf{P}_{1,t} \|_{\text{Fro}}^2 + \| \mathbf{P}_{2,t} \|_{\text{Fro}}^2 + \| \mathbf{P}_{3,t} \|_{\text{Fro}}^2. \tag{3.72}$$

Since the closed-form expression (3.51) represents an eigenvalue decomposition with orthonormal eigenvectors $\frac{\mathbf{v}_i\otimes\mathbf{v}_j+\mathbf{v}_j\otimes\mathbf{v}_i}{\sqrt{2}}$, $1\le i\le j\le n$, we have:

$$\mathbf{C}_P = \sum_{1\le i\le j\le n} 2(b_{j,i}1_{i\le r<j} + d_{i,j}1_{i<j\le r})\frac{(\mathbf{v}_i\otimes\mathbf{v}_j+\mathbf{v}_j\otimes\mathbf{v}_i)}{\sqrt{2}}\frac{(\mathbf{v}_i\otimes\mathbf{v}_j+\mathbf{v}_j\otimes\mathbf{v}_i)^T}{\sqrt{2}}, \tag{3.73}$$

and therefore, the three mean square error terms in the right-hand side of (3.72) read respectively, for each of the four algorithms

$$\mathrm{E}\| \mathbf{P}_{1,t} \|_{\text{Fro}}^2 = 2\gamma \sum_{1\le i<j\le r} d_{i,j} + o(\gamma), \tag{3.74}$$

$$\mathrm{E}\| \mathbf{P}_{2,t} \|^2_{\mathrm{Fro}} = 2\gamma \sum_{1\leq i\leq r<j\leq n} b_{j,i} + o(\gamma), \tag{3.75}$$

$$\mathrm{E}\| \mathbf{P}_{3,t} \|^2_{\mathrm{Fro}} = o(\gamma). \tag{3.76}$$

Note that (3.74) reduces to $o(\gamma)$ in case of the SGA algorithm. Note also that the formula (3.74) [resp., (3.75)] is equal to the first [resp., second] term of (3.66) (3.67) (3.68) (3.69), depending on the considered algorithm. As for the OFA algorithm, these two sommations are respectively over $n-r+1 \leq i < j \leq n$ and $1 \leq j < n-r+1 \leq i \leq n$. We note that our first order performance analysis cannot determine an equivalent expression for the deviation of orthonormality $\mathrm{E}\| \mathbf{W}_t^T \mathbf{W}_t - \mathbf{I}_r \|^2_{\mathrm{Fro}}$. We show in Section 4, that this MSE is (to the first order) proportional to γ for the GHA and OFA algorithms and to γ^2 for the WSA and SGA algorithm.

4 Simulations

We consider throughout this Section, the case $n = 4$, $r = 2$ associated with $\mathbf{R}_x = \mathrm{Diag}(1.75, 1.5, 0.5, 0.25)$. Clearly, the eigenvalues of $\mathbf{R}_x$ are 1.75, 1.5, 0.5 and 0.25 and the associated eigenvectors are $\mathbf{e}_i^n, i = 1, \ldots, 4$. The entries of the initial value $\mathbf{W}_0$ are chosen randomly uniformly on [0,1], then $\mathbf{w}_{0,k}, k = 1, \ldots, 4$ are normalized and all the learning curves averaged over 100 independent runs.

First of all, in order to compare the different algorithms studied, we consider the parameterized algorithms only. Fig. 1 and Fig. 2 show the learning curves of the mean square error of $\mathbf{W}_t$ for the SGA and WSA algorithms and of $\mathbf{P}_t$ for the SGA, WSA, and OFA algorithms respectively. We note that the choice $n-r = r$ allows us to compare the mean square errors of $\mathbf{P}_t$ for minorant and majorant algorithms. For the different algorithms, the step size γ is chosen so as to provide the same value for, respectively, $\gamma \mathrm{Tr}(\mathbf{C}_W)$ and $\gamma \mathrm{Tr}(\mathbf{C}_P)$. We select the values $\alpha_2 = 1$ and $\frac{\beta_2}{\beta_1} = 0.6$ for estimating eigenvectors and $\alpha_2 = 2$, $\frac{\beta_2}{\beta_1} = 0.9$ and $\beta = 5$ for estimating projection matrices associated with the faster speed of convergence.

For these parameters, Fig. 3, [resp. 4] shows the learning curves of the mean square error of $\mathbf{W}_t$ [resp., $\mathbf{P}_t$] and the learning curves of the associated deviation from orthonormality for these algorithms. We see that the SGA algorithm is the fastest for estimating both eigenvectors and projection matrices.

Fig. 5 shows the ratio of the estimated mean square error $\mathrm{E}\| \mathbf{P}_t - \mathbf{P}_* \|^2_{\mathrm{Fro}}$ over the theoretical asymptotic mean square error $\gamma \mathrm{Tr}(\mathbf{C}_P)$, as a function of γ, for the different algorithms studied. Our present asymptotic analysis is seen to be valid over a large range of γ ($\gamma < 0.01$), and the domain of "stability" is $\gamma < 0.035$ for which this ratio stays close to 1. This result supports our conjecture that the asymptotic covariance matrices of our recursive eigenvectors estimators are identical to the covariance matrices in the limiting distributions.

Finally, Fig. 6 shows that the deviation from orthonormality $d^2(\gamma) \stackrel{\mathrm{def}}{=} \mathrm{E}\| \mathbf{W}_t^T \mathbf{W}_t - \mathbf{I}_r \|^2_{\mathrm{Fro}}$ is proportional to γ [resp., to γ^2] in the domain of validity of (3.61) for the GHA and OFA algorithms [resp., for the WSA, SGA and Yang algorithms [8]], because $\log_{10} d^2(\gamma) = \log_{10} c + \alpha \log_{10} \gamma$ with $\alpha = 1$ or $\alpha = 2$.

5 Conclusion

In this paper, we have derived closed-form expressions of the covariance in distribution of the estimators of eigenvectors and of the associated projection matrix used in some adaptive gradient-like algorithms introduced in the neural network literature, after presenting these algorithms in a common framework. The asymptotic performances of these algorithms have been studied and closed-form expressions for the MSE, simulations for the convergence speed and the deviation from orthonormality have been derived. These results should prove useful in selecting the best algorithm for a given application and may also serve to popularize such algorithms in the signal processing community.

A Proof of theorem 1

If $\mathbf{U} \stackrel{\mathrm{def}}{=} \mathrm{Diag}(\mathbf{V}, \ldots, \mathbf{V})$ denotes the $nr \times nr$ block diagonal orthonormal matrix with $n \times n$ block diagonal $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, the $nr \times nr$ matrix $\mathbf{D}$ of the SGA, GHA and OFA algorithms can be written as

$$\mathbf{D} = \mathbf{U} \boldsymbol{\Delta} \mathbf{U}^T \tag{A.1}$$

where $\mathbf{\Delta}$ is an $nr \times nr$ triangular matrix. From (3.23),(3.24) and (3.26), the diagonal entries of the $n \times n$ diagonal blocks $\mathbf{\Delta}_{i,i}$ of $\mathbf{\Delta}$ are, for the SGA, GHA and OFA algorithms respectively

$$\begin{aligned}
(\mathbf{\Delta}_{i,i})_{k,k} &= -\alpha_i[(\lambda_i + \tfrac{\alpha_k}{\alpha_i}\lambda_k)1_{k<i} + 2\lambda_i 1_{k=i} + (\lambda_i - \lambda_k)1_{k>i}], && k = 1,\ldots,n \quad i = 1,\ldots,r \\
&= -[(\lambda_i 1_{k<i} + 2\lambda_i 1_{k=i} + (\lambda_i - \lambda_k)1_{k>i}], && k = 1,\ldots,n \quad i = 1,\ldots,r \\
&= (\lambda_i - \lambda_k)1_{k<i} + 2(\lambda_i - 1)1_{k=i} + (\lambda_i - (1+\beta)\lambda_k)1_{k>i}, && k = 1,\ldots,n \;\; i = n-r+1,\ldots,n.
\end{aligned} \tag{A.2}$$

Thanks to the decreasing order of the eigenvalues λ_i, and to (2.14), $(\mathbf{\Delta}_{i,i})_{k,k} < 0$, and thus, the eigenvalues of $\mathbf{D}$ are strictly negative real.

As for the WSA algorithm, $\mathbf{\Delta}$ is no longer a triangular matrix; rather, $\mathbf{\Delta} = \mathbf{\Delta}^1 + \mathbf{\Delta}^2$, where $\mathbf{\Delta}^1$ is a diagonal matrix. The diagonal entries of the $n \times n$ diagonal blocks $\mathbf{\Delta}^1_{i,i}$ of $\mathbf{\Delta}^1$ are

$$(\mathbf{\Delta}^1_{i,i})_{k,k} = -[(\lambda_i - (1 - \frac{\beta_i}{\beta_k}))1_{k<r,k\neq i} + 2\lambda_i 1_{k=i} + (\lambda_i - \lambda_k)1_{k>r}], \qquad k = 1,\ldots,n. \tag{A.3}$$

As for the block matrix $\mathbf{\Delta}^2$, its (i,j) block has all its entries equal to zero, except the entry at the position (j,i)

$$\mathbf{\Delta}^2_{i,j} = -\lambda_i \frac{\beta_i}{\beta_j} \mathbf{e}^r_j \mathbf{e}^{r^T}_i. \tag{A.4}$$

Consider the $nr \times nr$ orthonormal matrix $\mathbf{U}'$, the columns of which come from a permutation of the columns of $\mathbf{U}$ such that $\mathbf{U}' \stackrel{\text{def}}{=} (\mathbf{U}'_1, \mathbf{U}'_2)$, where $\mathbf{U}'_1$ is the $nr \times (n-r+1)r$ block diagonal matrix $\text{Diag}(\mathbf{V}_1,\ldots,\mathbf{V}_r)$, with $\mathbf{V}_i \stackrel{\text{def}}{=} (\mathbf{v}_i, \mathbf{v}_{r+1},\ldots,\mathbf{v}_n)$, and where $\mathbf{U}'_2$ is the $nr \times (r-1)r$ block matrix made of the $\frac{r(r-1)}{2}$ matrices $nr \times 2$ $(\mathbf{e}^r_i \otimes \mathbf{v}_j, \mathbf{e}^r_j \otimes \mathbf{v}_i)$ for all pairs (i,j) such that $1 \leq i < j \leq r$. We note that the particular ordering of these pairs is irrelevant for the following. Therefore

$$\mathbf{D} = \mathbf{U}'\mathbf{\Delta}'\mathbf{U}'^T \tag{A.5}$$

with

$$\mathbf{\Delta}' = \text{Diag}(\mathbf{\Delta}'_1, \mathbf{\Delta}'_2). \tag{A.6}$$

In (A.6), $\mathbf{\Delta}'_1$ is the $(n-r+1)r \times (n-r+1)r$ block diagonal matrix

$$\mathbf{\Delta}'_1 = \text{Diag}(\mathbf{\Delta}'^1_1,\ldots,\mathbf{\Delta}'^r_1) \tag{A.7}$$

with

$$\mathbf{\Delta}_1'^i = \mathrm{Diag}(-2\lambda_i, -(\lambda_i - \lambda_{r+1}), \ldots, -(\lambda_i - \lambda_n)) \tag{A.8}$$

and $\mathbf{\Delta}_2'$ is the $(r-1)r \times (r-1)r$ block diagonal matrix

$$\mathbf{\Delta}_2' = \mathrm{Diag}(\ldots, \mathbf{\Delta}_2'^{i,j}, \ldots) \tag{A.9}$$

with

$$\mathbf{\Delta}_2'^{i,j} = -\begin{pmatrix} \lambda_i - (1-a_{i,j})\lambda_j & \lambda_i a_{i,j} \\ \lambda_j a_{i,j}^{-1} & \lambda_j - (1-a_{i,j}^{-1})\lambda_i \end{pmatrix} \quad \text{for} \quad 1 \le i < j \le r, \tag{A.10}$$

and with $a_{i,j} \stackrel{\text{def}}{=} \frac{\beta_i}{\beta_j} < 1$. If $x'_{i,j}$ and $x''_{i,j}$ denote the eigenvalues of $\mathbf{\Delta}_2'^{i,j}$, it is straightforward to see that

$$x'_{i,j}x''_{i,j} = (\lambda_i - \lambda_j)((a_{i,j}^{-1} - 1)\lambda_i - (a_{i,j} - 1)\lambda_j) > 0 \qquad \text{and} \qquad x'_{i,j} + x''_{i,j} = -(a_{i,j}\lambda_j + a_{i,j}^{-1}\lambda_i) < 0,$$

so that $x'_{i,j}$ and $x''_{i,j}$ are either strictly negative real or conjugate complex with strictly negative real part. Last, observing, that the eigenvalues of $\mathbf{D}$ are those of $\mathbf{\Delta}_1'^i$ and $\mathbf{\Delta}_2'^{i,j}$, theorem 1 is proved.

B Proof of theorem 2

A closed-form expression of $\mathbf{C}_W$ can be given thanks to the previous change of basis. With the orthonormal basis $\mathbf{U}'$ [1] defined in Theorem 1, we have:

$$\mathbf{D} = \mathbf{U}'\mathbf{\Delta}'\mathbf{U}'^T, \qquad \mathbf{G} = \mathbf{U}'\mathbf{\Gamma}'\mathbf{U}'^T \quad \text{and} \quad \mathbf{C}_W = \mathbf{U}'\mathbf{\Sigma}'\mathbf{U}'^T. \tag{B.1}$$

On the basis $\mathbf{U}$', (3.16) reads

$$\mathbf{\Delta}'\mathbf{\Sigma}' + \mathbf{\Sigma}'\mathbf{\Delta}'^T + \mathbf{\Gamma}' = \mathbf{O} \tag{B.2}$$

where $\mathbf{\Delta}'$ and $\mathbf{\Gamma}'$ have the same structure as above [see eq. (A.6), (A.7) and (A.9)]. More precisely, the matrices $\mathbf{\Delta}_1'^i$ and $\mathbf{\Delta}_2'^{i,j}$ for the SGA, GHA, and OFA algorithms read, respectively

$$\mathbf{\Delta}_1'^i \;=\; \mathrm{Diag}(-2\alpha_i\lambda_i, -\alpha_i(\lambda_i - \lambda_{r+1}), \ldots, -\alpha_i(\lambda_i - \lambda_n)) \qquad i = 1, \ldots, r \tag{B.3}$$

[1] for the OFA algorithm, $\mathbf{U}_1' \stackrel{\text{def}}{=} \mathrm{Diag}(\mathbf{V}_{n-r+1}, \ldots, \mathbf{V}_n)$ with $\mathbf{V}_i \stackrel{\text{def}}{=} (\mathbf{v}_i, \mathbf{v}_1, \ldots, \mathbf{v}_{n-r})$ and $\mathbf{U_2}'$ is the $nr \times (r-1)r$ block matrix made of the $\frac{r(r-1)}{2}$ matrices $nr \times 2$ $(\mathbf{e}_i^r \otimes \mathbf{v}_j, \mathbf{e}_j^r \otimes \mathbf{v}_i)$ for all pairs (i,j) such that $n - r + 1 \le i < j \le n$.

$$\boldsymbol{\Delta}_2'^{i,j} = -\begin{pmatrix} \alpha_i(\lambda_i-\lambda_j) & 0 \\ \alpha_j(1+\frac{\alpha_i}{\alpha_j})\lambda_j & \alpha_j(\lambda_j+\frac{\alpha_i}{\alpha_j}\lambda_i) \end{pmatrix} \qquad 1\le i<j\le r \tag{B.4}$$

$$\boldsymbol{\Delta}_1'^{i} = \mathrm{Diag}[-2\lambda_i, -(\lambda_i-\lambda_{r+1}), \ldots, -(\lambda_i-\lambda_n)] \qquad i=1,\ldots,r \tag{B.5}$$

$$\boldsymbol{\Delta}_2'^{i,j} = -\begin{pmatrix} \lambda_i-\lambda_j & 0 \\ \lambda_j & \lambda_j \end{pmatrix} \qquad 1\le i<j\le r \tag{B.6}$$

$$\boldsymbol{\Delta}_1'^{i} = \mathrm{Diag}[(\lambda_i-\lambda_1), \ldots, (\lambda_i-\lambda_{n-r}), 2(\lambda_i-1)] \qquad i=n-r+1,\ldots,n \tag{B.7}$$

$$\boldsymbol{\Delta}_2'^{i,j} = \begin{pmatrix} \lambda_i-(1+\beta)\lambda_j & -\beta\lambda_i \\ 0 & \lambda_j-\lambda_i \end{pmatrix} \qquad n-r+1\le i<j\le n \tag{B.8}$$

and for the SGA, GHA, OFA and WSA algorithms, the matrices $\boldsymbol{\Gamma}_1'^{i}$ and $\boldsymbol{\Gamma}_2'^{i,j}$ read, respectively

$$\boldsymbol{\Gamma}_1'^{i} = \mathrm{Diag}(0, \alpha_i^2\lambda_i\lambda_{r+1}, \ldots, \alpha_i^2\lambda_i\lambda_n) \qquad i=1,\ldots,r \tag{B.9}$$

$$\boldsymbol{\Gamma}_2'^{i,j} = \alpha_i^2\lambda_i\lambda_j \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \qquad 1\le i<j\le r \tag{B.10}$$

$$\boldsymbol{\Gamma}_1'^{i} = \mathrm{Diag}(0, \lambda_i\lambda_{r+1}, \ldots, \lambda_i\lambda_n) \qquad i=1,\ldots,r \tag{B.11}$$

$$\boldsymbol{\Gamma}_2'^{i,j} = \lambda_i\lambda_j \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad 1\le i<j\le r \tag{B.12}$$

$$\boldsymbol{\Gamma}_1'^{i} = \mathrm{Diag}(\lambda_i\lambda_1, \ldots, \lambda_i\lambda_{n-r}, 0) \qquad i=n-r+1,\ldots,n \tag{B.13}$$

$$\boldsymbol{\Gamma}_2'^{i,j} = \lambda_i\lambda_j \begin{pmatrix} (1+\beta)^2 & (1+\beta) \\ (1+\beta) & 1 \end{pmatrix} \qquad n-r+1\le i<j\le n \tag{B.14}$$

$$\boldsymbol{\Gamma}_1'^{i} = \mathrm{Diag}(0, \lambda_i\lambda_{r+1}, \ldots, \lambda_i\lambda_n) \qquad i=1,\ldots,r \tag{B.15}$$

$$\boldsymbol{\Gamma}_2'^{i,j} = \lambda_i\lambda_j \begin{pmatrix} (1-a_{i,j})^2 & (1-a_{i,j})(1-a_{i,j}^{-1}) \\ (1-a_{i,j})(1-a_{i,j}^{-1}) & (1-a_{i,j}^{-1})^2 \end{pmatrix} \qquad 1\le i<j\le r \tag{B.16}$$

Thus, the unique symmetric solution $\boldsymbol{\Sigma}'$ of (B.2) is in the same form as $\boldsymbol{\Delta}'$ and $\boldsymbol{\Gamma}'$, and (B.2) reduces to uncoupled 1-D and 2-D Lyapunov equations. Therefore

$$\boldsymbol{\Sigma}' = \mathrm{Diag}(\boldsymbol{\Sigma}_1', \boldsymbol{\Sigma}_2') \tag{B.17}$$

where $\boldsymbol{\Sigma}_1'$ is the $(n-r+1)r \times (n-r+1)r$ block diagonal matrix

$$\boldsymbol{\Sigma}_1' = \mathrm{Diag}(\boldsymbol{\Sigma}_1'^{1}, \ldots, \boldsymbol{\Sigma}_1'^{r}), \quad [\text{resp., } \mathrm{Diag}(\boldsymbol{\Sigma}_1'^{n-r+1}, \ldots, \boldsymbol{\Sigma}_1'^{n}), \text{ for the OFA algorithm}] \tag{B.18}$$

and $\mathbf{\Sigma}'_2$ is the $(r-1)r \times (r-1)r$ block diagonal matrix

$$\mathbf{\Sigma}'_2 = \mathrm{Diag}(\ldots, \mathbf{\Sigma}_2'^{i,j}, \ldots) \tag{B.19}$$

with $\mathbf{\Sigma}'_1$ and $\mathbf{\Sigma}'_2$ solutions of uncoupled 1-D and 2-D Lyapunov equations, respectively. In particular, $\mathbf{\Sigma}'_1$ is the unique symmetric solution of the diagonal $n-r+1$-dimensional Lyapunov equation:

$$\mathbf{\Delta}_1'^i \mathbf{\Sigma}_1'^i + \mathbf{\Sigma}_1'^i \mathbf{\Delta}_1'^{iT} + \mathbf{\Gamma}_1'^i = \mathbf{O}. \tag{B.20}$$

Thus, making use of (B.9)-(B.3), (B.11)-(B.5), (B.13)-(B.7), and (B.15)-(A.8), $\mathbf{\Sigma}_1'^i$ reads respectively for the SGA, GHA, OFA and WSA algorithms

$$\begin{aligned}
\mathbf{\Sigma}_1'^i &= \mathrm{Diag}[0, \alpha_i \frac{\lambda_i \lambda_{r+1}}{2(\lambda_i - \lambda_{r+1})}, \ldots, \alpha_i \frac{\lambda_i \lambda_n}{2(\lambda_i - \lambda_n]}) \qquad i = 1, \ldots, r & \text{(B.21)} \\
&= \mathrm{Diag}[0, \frac{\lambda_i \lambda_{r+1}}{2(\lambda_i - \lambda_{r+1})}, \ldots, \frac{\lambda_i \lambda_n}{2(\lambda_i - \lambda_n)}] \qquad i = 1, \ldots, r & \text{(B.22)} \\
&= \mathrm{Diag}[\frac{\lambda_i \lambda_1}{2(\lambda_1 - \lambda_i)}, \ldots, \frac{\lambda_i \lambda_{n-r}}{2(\lambda_{n-r} - \lambda_i)}, 0] \qquad i = n-r+1, \ldots, n & \text{(B.23)} \\
&= \mathrm{Diag}[0, \frac{\lambda_i \lambda_{r+1}}{2(\lambda_i - \lambda_{r+1})}, \ldots, \frac{\lambda_i \lambda_n}{2(\lambda_i - \lambda_n)}] \qquad i = 1, \ldots, r & \text{(B.24)}
\end{aligned}$$

and $\mathbf{\Sigma}_2'^{i,j}$ is the unique symmetric solution of the triangular 2-D Lyapunov equation:

$$\mathbf{\Delta}_2'^{i,j} \mathbf{\Sigma}_2'^{i,j} + \mathbf{\Sigma}_2'^{i,j} \mathbf{\Delta}_2'^{i,jT} + \mathbf{\Gamma}_2'^{i,j} = \mathbf{O} \tag{B.25}$$

the solution of which, thanks to (B.4), (B.10), (B.6), (B.12) and (A.10), (B.16), in case of the SGA, GHA and OFA algorithms, is respectively

$$\begin{aligned}
\mathbf{\Sigma}_2'^{i,j} &= \frac{\alpha_i \lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} & \text{(B.26)} \\
&= \frac{\lambda_j}{2(\lambda_i - \lambda_j)} \begin{bmatrix} \lambda_i & -\lambda_j \\ -\lambda_j & \lambda_j \end{bmatrix} & \text{(B.27)} \\
&= \begin{bmatrix} g^{ij} & h^{ij} \\ h^{ij} & \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} \end{bmatrix} & \text{(B.28)}
\end{aligned}$$

with $g^{ij} \stackrel{\text{def}}{=} -\frac{\lambda_i\lambda_j}{2(\lambda_i-(1+\beta)\lambda_j)}[(1+\beta)^2 - \frac{2\lambda_i}{\lambda_j}(1+\beta+\frac{\beta\lambda_i}{2(\lambda_j-\lambda_i)})]$ and $h^{ij} \stackrel{\text{def}}{=} \frac{\lambda_i}{\beta}(1+\beta-\frac{\beta\lambda_i}{2(\lambda_i-\lambda_j)})$. As for the WSA algorithm, since (B.25) is no longer triangular, $\mathbf{\Sigma}_2'^{i,j}$ reads after some tedious calculus

$$\mathbf{\Sigma}_2'^{i,j} = \lambda_i\lambda_j \begin{bmatrix} b^{ij} & d^{ij} \\ d^{ij} & c^{ij} \end{bmatrix}, \tag{B.29}$$

with

$$b^{ij} = \frac{2\lambda_i a_{i,j} d^{ij} - (1-a_{i,j})^2}{2(\lambda_j(1-a_{i,j})-\lambda_i)}, \qquad c^{ij} = \frac{2\lambda_j a_{i,j}^{-1} d^{ij} - (1-a_{i,j}^{-1})^2}{2(\lambda_i(1-a_{i,j}^{-1})-\lambda_j)}, \tag{B.30}$$

and with $d^{ij} \stackrel{\text{def}}{=} -\frac{(1-a_{i,j})^2 a_{i,j}^{-1}(\lambda_i^2(2a_{i,j}^{-1}-1)+\lambda_j^2(2a_{i,j}-1)+\lambda_i\lambda_j(4-a_{i,j}-a_{i,j}^{-1}))}{2(\lambda_j a_{i,j}+\lambda_i a_{i,j}^{-1})(\lambda_i^2(a_{i,j}^{-1}-1)+\lambda_j^2(a_{i,j}-1)+\lambda_i\lambda_j(2-a_{i,j}-a_{i,j}^{-1}))}$.

Putting together the results (B.21)-(B.26), (B.22)-(B.27), (B.23)-(B.28) and (B.24)-(B.29), and thanks to (B.1), Theorem 2 is proved.

References

[1] E. Moulines, P. Duhamel, J.F. Cardoso, S. Mayrargue, "Subspace methods for blind identification of multichannel FIR filters," *IEEE Trans. Signal Processing*, vol. 43, no. 2, pp. 516-525, Feb. 1995.

[2] E. Oja, "Principal components, minor components and linear neural networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.

[3] J. Dehaene, *Continuous-time matrix algorithms, systolic algorithms and adaptive neural networks*, Ph.D. dissertation, Katholieke Univ. Leuven, Belgium, Oct. 1995.

[4] P. Comon, G.H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327-1343, Aug. 1990.

[5] C.M. Kuan, K. Hornik, "Convergence of learning algorithms with constant learning rates," *IEEE Trans. Neural Networks*, vol. 2, no. 5, pp. 484-489, Sept. 1991.

[6] B. Yang, F. Gersemsky, "Asymptotic distribution of recursive subspace estimators," *Proc. ICASSP* Atlanta, GA, May 1996, pp. 1764-1767.

[7] J.P. Delmas, "Performance analysis of parametrized adaptive eigensubpace algorithms," in *Proc. ICASSP* Detroit, MI, May 1995, pp. 2056-2059.

[8] J.P. Delmas, J.F. Cardoso "Performance analysis of an adaptative algorithm for tracking dominant subspaces," submitted to *IEEE Trans. on Signal Processing*, Dec. 1996.

[9] H. Rutishauser, "Computational aspects of F.L.Bauer's simultaneous iteration method," *Numer. Math.* vol. 13 pp. 4-13, 1969.

[10] R. Williams, "Feature discovery through error-correcting learning," Inst. Cognitive Sci., Univ. California, San Diego, Tech. Rep. 8501, 1985.

[11] W-Y. Yan, U. Helmke, J.B. Moore, "Global analysis of Oja's flow for neural networks," *IEEE Trans. on Neural Networks*, vol. 5, no. 5, pp. 674-683, Sept. 1994.

[12] B. Yang, "Projection approximation subspace tracking", *IEEE, Trans. Signal Processing*, vol. 43, no. 1, pp. 95-107, Jan. 1995.

[13] B. Yang, "Convergence analysis of the subspace tracking algorithms PAST and PASTd," in *Proc. ICASSP* Atlanta, GA, May 1996, pp. 1760-1763.

[14] E. Oja, *Subspace methods of pattern recognition*, Letchworth, England, Research Studies Press and John Wiley and Sons, 1983.

[15] T.D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward network," *Neural Networks*, vol. 2, pp. 459-473, 1989.

[16] E. Oja, H. Ogawa and J. Wangviwattana "Principal component analysis by homogeneous neural networks, Part I: The weighted subspace criterion," *IEICE Trans. Inform. and Syst.*, vol.E75-D, pp. 366-375, 1992.

[17] E. Oja, H. Ogawa and J. Wangviwattana "Principal component analysis by homogeneous neural networks, Part II: Analysis and extensions of the learning algorithms," *IEICE Trans. Inform. Syst.*, vol.E75-D, pp. 376-382, 1992.

[18] E. Oja, "A simplified neuron model as a principal components analyzer," *J. Math. Biol.*, vol. 15, pp. 267-273, 1982.

[19] C. Riou, T. Chonavel, P.Y. Cochet, "Adaptive subspace estimation - Application to moving sources localization and blind channel identification," in *Proc. ICASSP* Atlanta, GA, May 1996, pp. 1649-1652.

[20] L. Xu, E. Oja, C. Suen, "Modified Hebbian learning for curve and surface fitting," *Neural Networks*, vol. 5, no.3, pp. 441-457, 1992.

[21] A. Benveniste, M. Métivier, P. Priouret, *Adaptive algorithms and stochastic approximations*, New York: Springer Verlag, 1990.

[22] N. Delfosse , P. Loubaton "Adaptive blind separation of independent sources: a deflation approach," *Signal Processing*, vol.45, pp. 59-83, 1995.

[23] K. Hornik, C.M. Kuan, "Convergence analysis of local feature extraction algorithms," *Neural Networks*, vol. 5, pp. 229-240, 1992.

[24] E. Oja, J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. anal. Applications*, vol.106, pp. 69-84, 1985.

[25] T.W. Anderson, *An introduction to multivariate statistical analysis.* 2nd ed., New York: Wiley, 1984.

[26] C.R. Rao, *Linear statistical inference and its applications*, New York: Wiley, 1973.

[27] D.R. Brillinger, *Times series, data analysis and theory.* San Francisco, CA: Holden-Day, 1980.

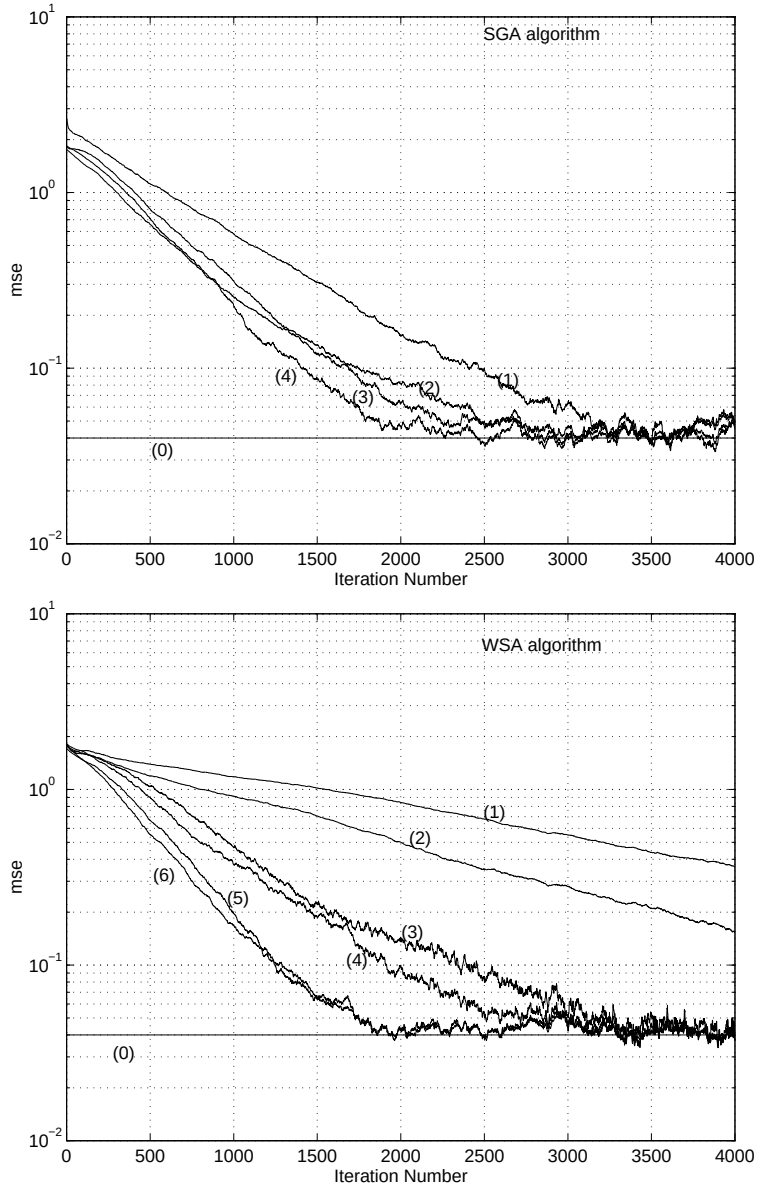

Figure 1: Learning curves of the mean square error $\mathrm{E}\| \mathbf{W}_t - \mathbf{W}_* \|^2_{\mathrm{Fro}}$ averaging 100 independent runs for respectively the SGA [resp. WSA] algorithm for different values of parameter $\alpha_2 = 10$ (1), 0.5 (2), 2 (3), 1 (4), [resp. $\frac{\beta_2}{\beta_1} = 0.96$ (1), 0.9 (2), 0.1 (3), 0.2 (4), 0.4 (5), 0.6 (6)], compared to $\gamma \mathrm{Tr}(\mathbf{C}_W)$ (0).

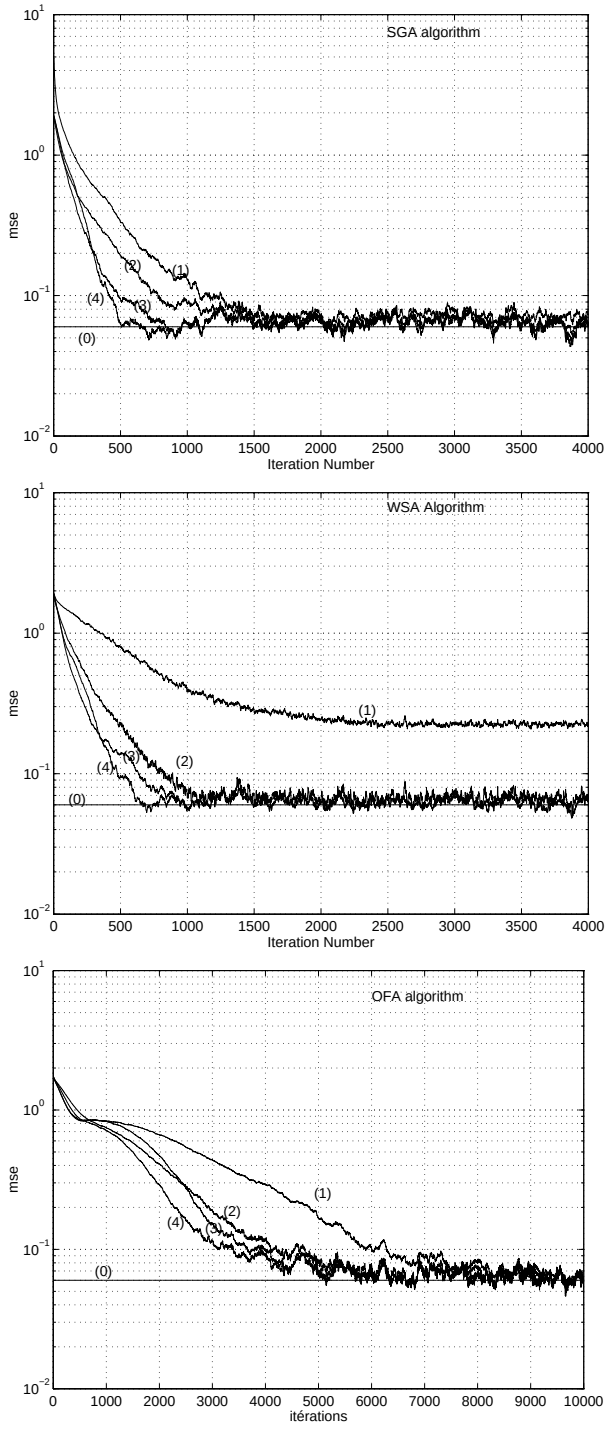

Figure 2: Learning curves of the mean square error $\mathrm{E}\| \mathbf{P}_t - \mathbf{P}_* \|^2_{\mathrm{Fro}}$ averaging 100 independent runs for respectively the SGA [resp. WSA and OFA] algorithms for different values of parameter $\alpha_2 = 10$ (1), 0.5 (2), 1 (3), 2 (4) [resp. $\frac{\beta_2}{\beta_1} = 0.1$ (1), 0.2 (2), 0.6 (3), 0.9 (4), $\beta = 2.5$ (1), 10 (2), 3 (3), 5 (4)], compared to $\gamma \mathrm{Tr}(\mathbf{C}_W)$ (0).

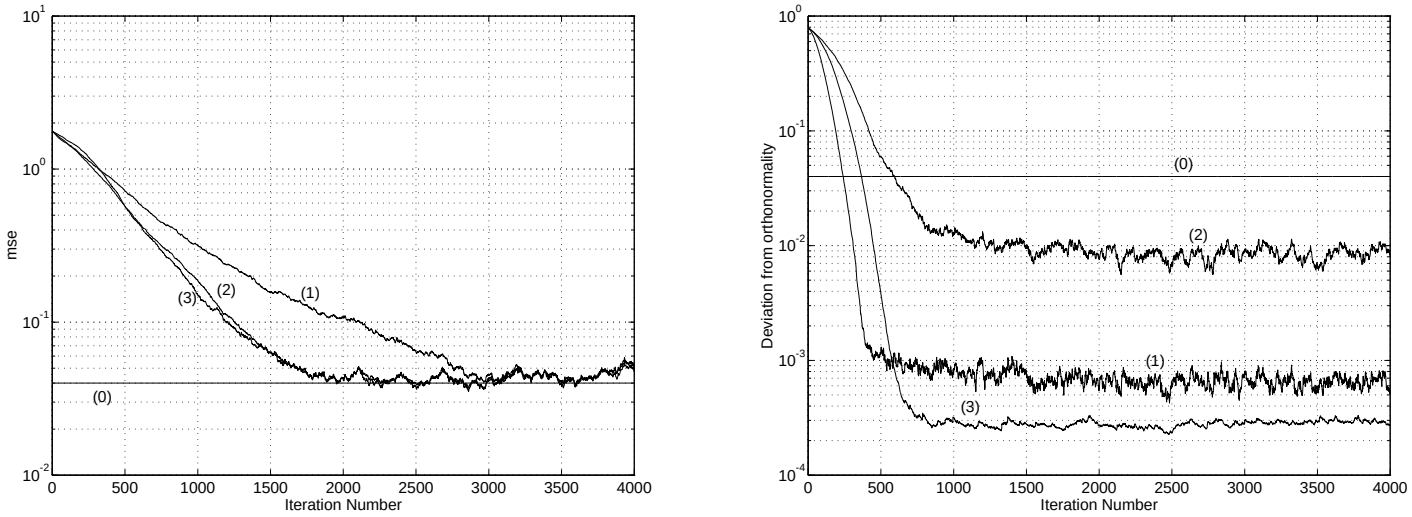

Figure 3: Learning curves of the mean square error $\mathrm{E}\| \mathbf{W}_t - \mathbf{W}_* \|^2_{\mathrm{Fro}}$ and deviation to orthonormality $\mathrm{E}\| \mathbf{W}_t^T \mathbf{W}_t - \mathbf{I}_r \|^2_{\mathrm{Fro}}$ averaging 100 independent runs for respectively WSA algorithm $\frac{\beta_2}{\beta_1} = 0.6$ (1), GHA algorithm (2) and SGA algorithm $\alpha_2 = 1$ (3), compared to $\gamma \mathrm{Tr}(\mathbf{C}_W)$ (0).

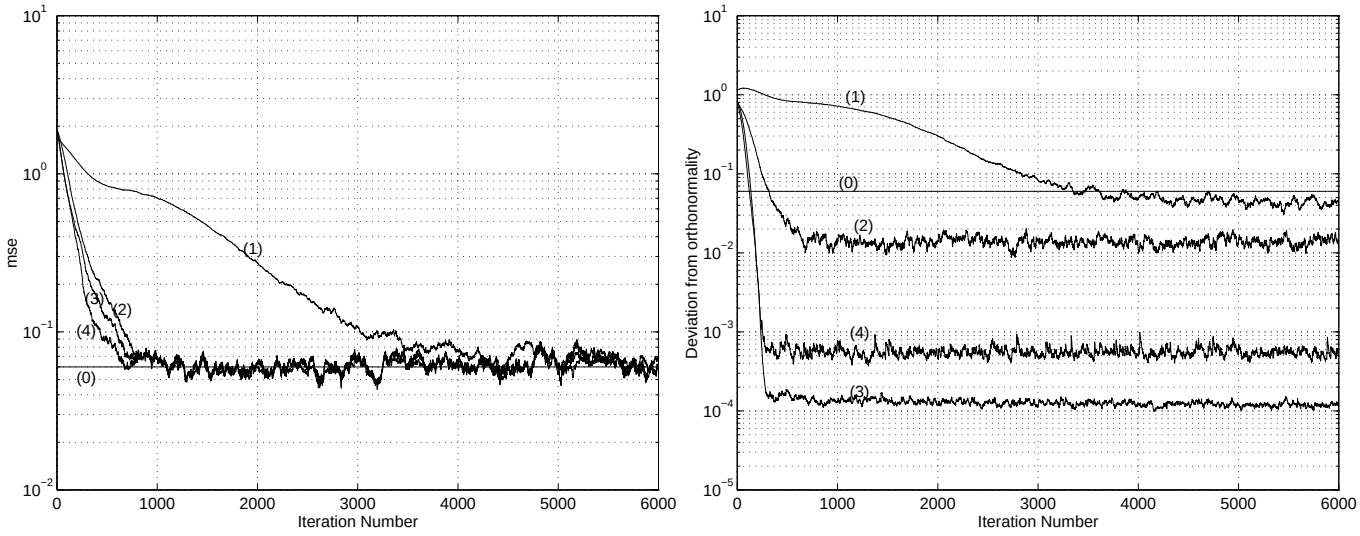

Figure 4: Learning curves of the mean square error $\mathrm{E}\| \mathbf{P}_t - \mathbf{P}_* \|^2_{\mathrm{Fro}}$ and deviation to orthonormality $\mathrm{E}\| \mathbf{W}_t^T \mathbf{W}_t - \mathbf{I}_r \|^2_{\mathrm{Fro}}$ averaging 100 independent runs for respectively OFA algorithm $\beta = 5$ (1), GHA algorithm (2), WSA algorithm $\frac{\beta_2}{\beta_1} = 0.9$ (3) SGA algorithm $\alpha_2 = 2$ (4), compared to $\gamma \mathrm{Tr}(\mathbf{C}_W)$ (0).

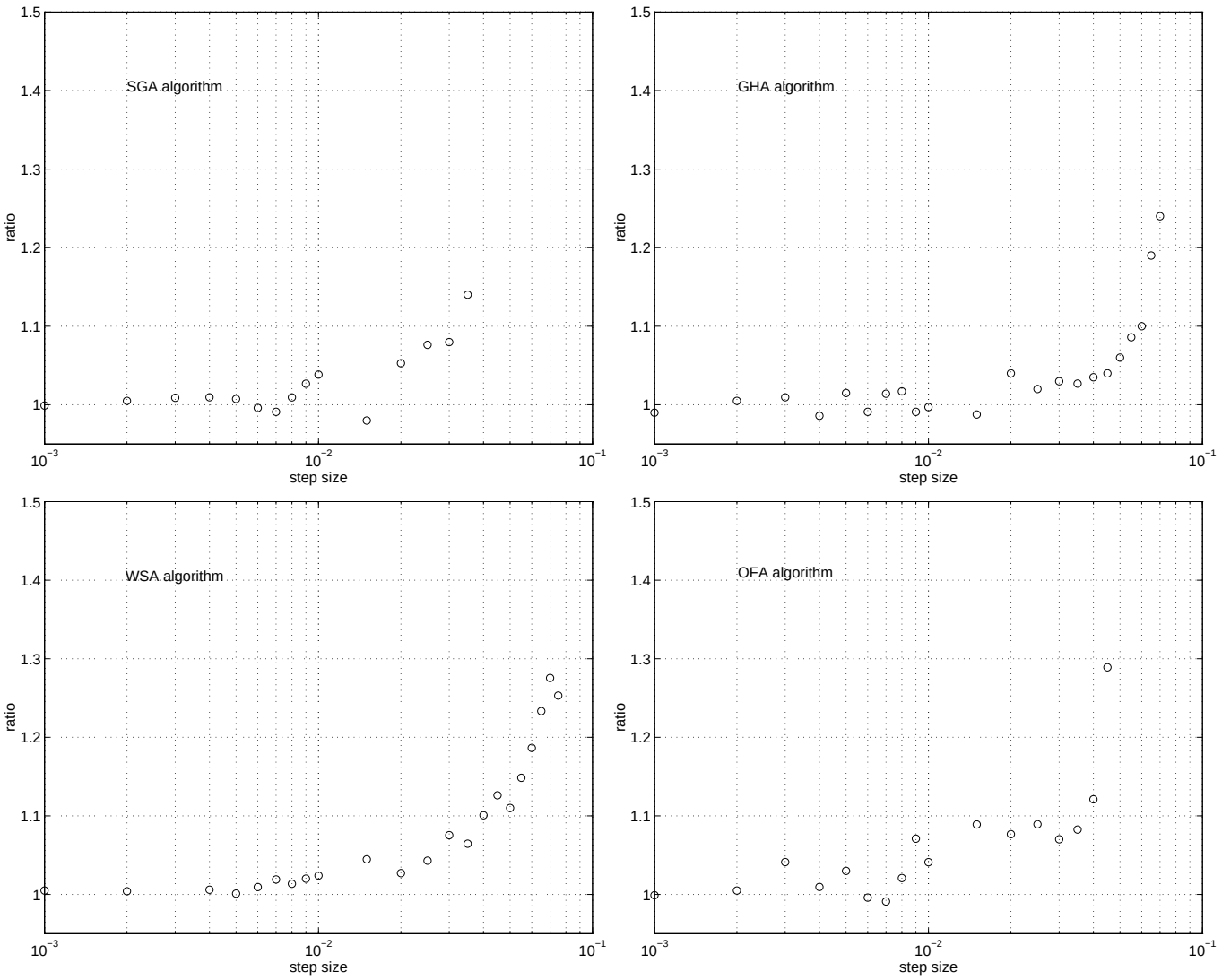

Figure 5: Ratio of the estimated mean square error $E\| \mathbf{P}_t - \mathbf{P}_* \|_{\text{Fro}}^2$ by averaging 400 independent runs to the theoretical asymptotic mean square error $\gamma \text{Tr}(\mathbf{C}_P)$ as a function of γ for the SGA algorithm $\alpha_2 = 2$, the GHA algorithm, the WSA algorithm $\frac{\beta_2}{\beta_1} = 0.9$ and the OFA algorithm $\beta = 5$.

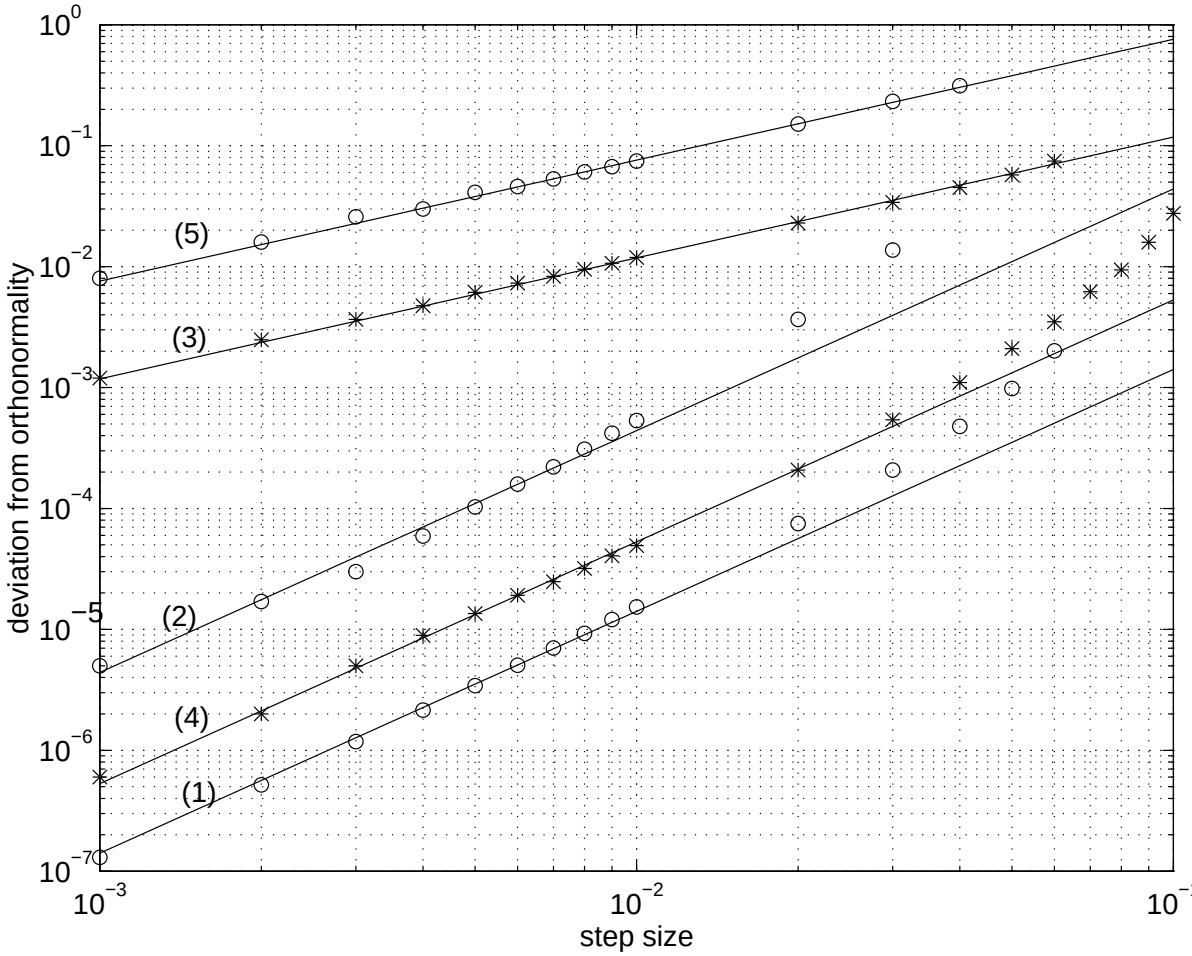

Figure 6: Deviation from orthonormality $d^2(\gamma) \stackrel{\text{def}}{=} \mathrm{E}\| \mathbf{W}_t^T \mathbf{W}_t - \mathbf{I}_r \|_{\mathrm{Fro}}^2$ at "convergence" estimated by averaging 100 independent runs as a function of γ in log-log scales for the Yang (1), SGA (2), GHA (3), WSA (4), OFA (5) algorithms.