A Neural Minor Component Analysis Approach to Robust Constrained Beamforming

Simone Fiori

Faculty of Engineering – University of Perugia, Italy
Via Pentima bassa, 21 – I-05100 Terni (Italy)

E-mail: sfr@unipg.it

February 9, 2003

Abstract

Since the pioneering work of Amari and Oja, principal component neural networks and their extensions have become an active adaptive signal processing research field. One of such extensions is minor component analysis (MCA), that proves to be effective in problems such as robust curve/surface fitting and noise reduction. The aims of this paper are to give a detailed and homogeneous review of one-unit first minor/principal component analysis and to propose an application to robust constrained beamforming. In particular, after a careful presentation of first/minor component analysis algorithms based on a single adaptive neuron, along with relevant convergence/steady-state theorems, it is shown how the adaptive robust constrained beamforming constrained theory by Cox *et al.* may be advantageously recast into an MCA seeting. Experimental results performed on a triangular array of microphones introduced in a teleconference context helps assessing the usefulness of the proposed theory.

Keywords: Artificial neural systems; Minor component analysis; Adaptive beamforming; Microphone array; Robust/constrained beamforming.

1 Introduction

Adaptive principal component analysis (PCA) by neural networks is a statistical signal processing technique, extensively investigated by Oja [40, 41, 42], which allows for extracting second-order features from a given random signal or a data-stream.

Likely, one of the reasons of the success of the neural PCA theory is its usefulness for solving many signal processing problems, as illustrated for instance in [17, 29, 37] and references therein, where extracting the first principal components is shown to be of prime importance. Nevertheless, it has been clearly shown that computing the *last* principal components of a data sequence, i.e. those principal components endowed with the smallest (non-zero) powers, may be very useful as well, for instance in moving target following [36], frequency estimation [39], adaptive array processing, emitter location and signal parameter estimation [48], biological data analysis and understanding [50], and noise reduction problems and function approximation like curve and surface fitting [3, 52]. The extraction of the last principal components is usually referred to as minor component analysis (MCA).

In this paper we consider the MCA learning rule proposed in [42] that allows to extract the first minor component from a stationary multivariate random process; we refer to Oja's work also because his formulation, based on the definition of a cost function to be minimized under right constraints, is very clear and allows us to penetrate the mathematical structure of the learning algorithm which we treat as a (non-linear, coupled) dynamical system to study. We first recall some preliminary results on first principal component analysis and pay special attention to the mathematical structure of FPCA learning system and to its convergence properties. Then we show that the direct extension of Oja's first principal component analysis rule to a first minor component analysis (FMCA) one is not possible, as the obtained learning algorithm comes unstable. Then we recall from [43] a simple but very interesting method for making this learning algorithm be stable, and formally prove that the stabilization theory is effective.

In the present contribution we present an application of FMCA to robust constrained beamforming.

Beamformers are frequently employed in acoustic applications to locate acoustic sources by microphone arrays without physical array-steering. Two or more microphones spaced apart form a microphone array. Such array, together with an adaptive beamforming program, can be used to implement a powerful directional listening device. Microphone array beamforming in acoustic systems relies on directionality to separate the desired speech from interfering noises. In acoustic beamforming, sounds coming from the direction of the speaking person are amplified, while sounds and disturbances coming from other directions are attenuated. For steady-state noises whose characteristics differ from speech, such as computer fan noise, air conditioner noise, automobile engine and road noise, it is possible to separate speech and attenuate noise using only one microphone. With the noise changing relatively to the speech, adaptive microphone array beamforming can be employed to enhance the speech signal. The system requires multiple analog-to-digital converters, one for each microphone.

Among other signal processing techniques, delay-and-sum beamformers have long been used to locate signal sources: As others steered beamformer based locators, delay-and-sum beamformers scan the area-of-interest with a beam in order to find the sector of space which yields the highest beam output power. However, the use of the delay-and-sum beamformers in conjunction to traditional analog-to-digital converters results in a memory intensive implementation and therefore it has been widely replaced by other beamforming algorithms.

Beamforming systems can be used to reduce noise in hearing aids, in teleconferencing systems, in hands-free microphones in automobiles and computer terminals, for speaker phones and speech recognition systems.

In teleconferencing systems, microphone arrays endowed with a beamforming algorithm provide a means for determining the point of sound origin: Speaker coordinates may be used to direct a camera at a member of an audience during a question-and-answer session, and can also provide directed sound capturing, thus a microphone array system may be used to eliminate the need for a human camera operator in an auditorium or conference hall environment.

In the present paper, we recall the microphone-array beamforming problem formulation of Cox *et al.* [14, 15] and show that it closely resembles an MCA optimization problem. On the basis of this finding, the MCA algorithm considered here is applied to acoustic beamforming problem and its numerical performances are illustrated.

2 First principal component and first minor component extraction

Let us consider the stationary multivariate random process $\mathbf{x}(t) \in \mathbb{R}^p$ and suppose it is zero-mean and its covariance matrix $\mathbf{\Phi}$ exists bounded; if $\mathbf{\Phi}$ is not diagonal, then the components of $\mathbf{x}(t)$ are statistically correlated and it would be of some use to find a linear operator $\mathbf{F}$ such that the new random signal defined by $\mathbf{y}(t) \stackrel{\text{def}}{=} \mathbf{F}\mathbf{x}(t) \in \mathbb{R}^m$ has uncorrelated components, with m being the smallest possible dimension such that $\mathbf{y}(t)$ represents $\mathbf{x}(t)$ with a representation error lower than a desired threshold [29, 34, 41, 51]. The operator $\mathbf{F}$ is known to be the matrix formed by the eigenvectors of $\mathbf{\Phi}$ corresponding to its largest eigenvalues [29]. Let σ_k be the power (eigenvalue) of the k^{th} component, and let y_i be the element of $\mathbf{y}(t)$ that has the greatest power: The $y_i(t)$ is termed *first principal component of* $\mathbf{x}(t)$; let y_j be the element of $\mathbf{y}(t)$ that has the second greatest power: The $y_j(t)$ is termed the second principal component of $\mathbf{x}(t)$, and so on.

Consider now a single linear neural unit. In 1982, Oja [40] proposed to use this simple neural unit to extract the first principal component from the input, that is to perform first principal component analysis (FPCA) of the input. Since this pioneering work and the previous studies on auto-association of Amari [2], several new learning algorithms have been proposed for extending the one-unit neural system to a complete neural network for the extraction of more than one principal component. Among others, contributions in this field have been given by Sanger [47], who used a on-line version of the well-known Gram-Schmidt orthogonalization algorithm, Rubner and Tavan [46] and Kung and Diamantaras [17] that introduced a linear neural network endowed with lateral inhibitory connections and an additional set of learning equations for achieving output decorrelation, further developed by the present author [23].

Over recent years, several authors tried to give different rules for generalizing classical ones. It is worth citing the successive application of modified Hebbian learning (SAMH) by Abbas and Fahmi [1], who introduced the concept of sequential extraction of principal components from previously deflated data; the recursive least square approach (RLS-PCA) by Bannour and Azimi-Sadjadi [5]; the cascade recursive least-squares approach (CRLS) by Cichocki *et al.* [12], that combines the advantages of both SAMH and RLS-PCA. A thorough comparison of these algorithms and of their performances on different classes of real-world data has recently appeared in [13].

It is also worth citing the non-linear extensions to PCA by Oja, Karhunen *et al.* [21, 33, 34, 49], the recently developed extensions to classical PCA to its complex-valued counterpart by DeCastro *et al.* [16] and non-linear complex-valued counterpart for performing independent component analysis of circularly-distributed signals by the present author [24, 26].

Consider the linear neural unit described by $y(t) = \mathbf{w}^T(t)\mathbf{x}(t)$, where $\mathbf{x} \in \mathbb{R}^p$ is the input vector, $\mathbf{w} \in \mathbb{R}^p$ represents the weight-vector, and y denotes the neuron's output, as depicted in Figure 1. Let us suppose that this unit is used for extracting the first principal component from the input random signal, that

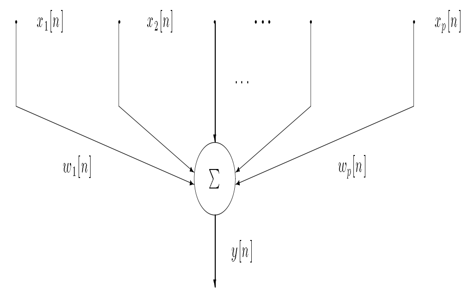

Figure 1: An exemplary linear neuron.

is $y(t)$ should represent $\mathbf{x}(t)$ in the best way, in the sense that the representation error $E_{\mathbf{x}}[||\mathbf{x}-y\mathbf{w}||^2|\mathbf{w}]$ should be minimized. Here $E_{\mathbf{x}}[\cdot|\mathbf{w}]$ denotes mathematical expectation with respect to $\mathbf{x}$ under the hypothesis $\mathbf{w}$. The problem may be expressed as:

$$\text{Solve}: \min_{\mathbf{w}\in\mathbb{R}^p}(E_{\mathbf{x}}[||\mathbf{x}||^2] - E_{\mathbf{x}}[y^2|\mathbf{w}]) \text{ under } \mathbf{w}^T\mathbf{w} = 1 \ . \tag{1}$$

Note that $E_{\mathbf{x}}[||\mathbf{x}||^2]$ does not depend on $\mathbf{w}$, thus we may simply try to maximize $E_{\mathbf{x}}[y^2|\mathbf{w}]$ under the constraint $\mathbf{w}^T\mathbf{w} = 1$. To this aim the following objective function may be considered [42]:

$$J(\mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{2}E_{\mathbf{x}}[y^2|\mathbf{w}] + \frac{\lambda}{2}(\mathbf{w}^T\mathbf{w} - 1) \ , \tag{2}$$

where $E_{\mathbf{x}}[y^2|\mathbf{w}]$ represents the power of the neuron's output and the additional term $\lambda(\mathbf{w}^T\mathbf{w} - 1)$ is used for enforcing the constraint $\mathbf{w}^T\mathbf{w} = 1$ by means of the Lagrange multiplier λ. The gradient of $J(\mathbf{w})$ computed with respect to $\mathbf{w}$ is found to be:

$$\frac{\partial J}{\partial \mathbf{w}} = E_{\mathbf{x}}[y\mathbf{x}|\mathbf{w}] + \lambda\mathbf{w} \ . \tag{3}$$

The optimal multiplier λ^{opt} may be found by solving the equation $\mathbf{w}^T\frac{\partial J}{\partial \mathbf{w}} = 0$ and recalling that optimality requires $||\mathbf{w}||^2 = 1$; this gives:

$$\mathbf{w}^T\frac{\partial J}{\partial \mathbf{w}} = E_{\mathbf{x}}[y^2|\mathbf{w}] + \lambda = 0 \ ,$$

that leads to $\lambda^{\text{opt}} = -E_{\mathbf{x}}[y^2|\mathbf{w}]$. Now the optimal steepest descent direction at $\mathbf{w}$ looks:

$$\left(\frac{\partial J}{\partial \mathbf{w}}\right)^{\text{opt}} = E_{\mathbf{x}}[y\mathbf{x}|\mathbf{w}] - E_{\mathbf{x}}[y^2|\mathbf{w}]\mathbf{w} \ , \tag{4}$$

thus we are able to define the steepest descent learning rule:

$$\frac{d\mathbf{w}}{dt} = \left(\frac{\partial J}{\partial \mathbf{w}}\right)^{\text{opt}} = E_{\mathbf{x}}[y\mathbf{x} - y^2\mathbf{w}|\mathbf{w}] \ , \tag{5}$$

that has been found and studied in [40] and discussed by many authors over the recent years (for a recent review, see e.g. [13]).

2.1 Convergence analysis of the first principal component analyzer

Let us suppose $\mathbf{x}(t)$ to be a zero-mean stationary random process with finite covariance; by defining the covariance matrix $\mathbf{\Phi} \stackrel{\text{def}}{=} E_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T]$, the differential equation (5) rewrites:

$$\begin{cases} \frac{d\mathbf{w}(t)}{dt} = \mathbf{\Phi}\mathbf{w}(t) - (\mathbf{w}^T(t)\mathbf{\Phi}\mathbf{w}(t))\mathbf{w}(t) \ , \\ \mathbf{w}(0) = \mathbf{w}_0 \ . \end{cases} \tag{6}$$

This dynamical system has a set of stationary points defined by:

$$\mathcal{E}_* \stackrel{\text{def}}{=} \{\mathbf{w} \in \mathbb{R}^p : \mathbf{\Phi}\mathbf{w} - \sigma\mathbf{w} = \mathbf{0}\} \ . \tag{7}$$

Except for the trivial solution $\mathbf{w} = \mathbf{0}$, the members $\mathbf{w} \in \mathcal{E}_*$ coincide to the eigenvectors of $\mathbf{\Phi}$ with eigenvalues σ.

Here we aim to prove the following Theorem, presented here in a new slant, stating the convergence of the system (6) to the eigenvector in $\mathcal{E}_*$ corresponding to the largest eigenvalue σ. The proof of this Theorem follows the one successfully used in [29, 47], and is very helpful to clarify the mathematical methods involved in the study of neural FPCA learning systems. It is given in Appendix A.1.

Theorem 1 *Suppose $\mathbf{\Phi} \in \mathbb{R}^{p\times p}$ is symmetric and positive-definite in (6) with eigenpairs $(\sigma_1, \mathbf{q}_1)$, $(\sigma_2, \mathbf{q}_2)$, ..., $(\sigma_p, \mathbf{q}_p)$. Suppose further that eigenvalues are distinct and arranged in descending order, eigenvectors are normalized so that $\mathbf{q}_k^T\mathbf{q}_k = 1$, and $\mathbf{w}_0^T\mathbf{q}_1 \neq 0$. Then there holds:*

$$\lim_{t\to+\infty} \mathbf{w}(t) = \pm\mathbf{q}_1 \ .$$

Note the importance of the condition $\mathbf{w}_0^T\mathbf{q}_1 \neq 0$, that clearly may not be explicitly fulfilled. In practice, this problem can be solved by randomly choosing $\mathbf{w}_0$; this makes the condition fulfilled 'with probability 1' [29]. Also, note that the hypothesis that $\mathbf{x}(t)$ is stationary may be partially relaxed by allowing instead the second-order statistics of $\mathbf{x}(t)$ to be slowly time-varying, namely $\mathbf{\Phi} = \mathbf{\Phi}(t)$. It also deserves to note that equation (26) describes the dynamics of the pseudo-modes and, in turn, the speed that the principal modes converge with to their asymptotic values; significantly, the speed is proportional to the differences $\sigma_h - \sigma_1$: The larger is the eigenvalues' spread, the faster is neurons' convergence. This quantitative explanation coincides to our intuitive understanding that the extraction of an eigenvector is easier when the components are well separated in eigenvalues.

2.2 First minor component analysis (FMCA) by the FPCA

By definition of minor component analysis, we have now to find the weight-vector $\mathbf{w}$ that *minimizes* the power $E_{\mathbf{x}}[y^2|\mathbf{w}]$ of the neuron's output. To extend the above first principal component analysis theory to a first minor component analysis theory in such a way is not a straightforward task: Particularly, it is not possible to replace maximization of the criterion (2) with its minimization, in that the resulting FMCA learning rule:

$$\begin{cases} \frac{d\mathbf{w}}{dt} = -\left(\frac{\partial J}{\partial \mathbf{w}}\right)^{\mathrm{opt}} = E_{\mathbf{x}}[-y\mathbf{x} + y^2\mathbf{w}|\mathbf{w}] \ , \\ \mathbf{w}(0) = \mathbf{w}_0 \ , \end{cases} \tag{8}$$

results to be *unstable*. This result has been discussed in [43, 52]. An alternative interpretation is given by the following Theorem whose proof finds in Appendix A.2.

Theorem 2 *Let $\mathbf{\Phi} \in \mathrm{I\!R}^{p\times p}$ be a symmetric and positive-definite matrix with eigenpairs $(\sigma_1, \mathbf{q}_1)$, $(\sigma_2, \mathbf{q}_2)$, ..., $(\sigma_p, \mathbf{q}_p)$. Suppose eigenvalues are distinct and arranged in descending order, eigenvectors are normalized so that $\mathbf{q}_k^T\mathbf{q}_k = 1$, and $\mathbf{w}_0^T\mathbf{q}_p \neq 0$. Then solution $\mathbf{w}(t)$ to the FMCA equation $\dot{\mathbf{w}} = -\mathbf{\Phi}\mathbf{w} + (\mathbf{w}^T\mathbf{\Phi}\mathbf{w})\mathbf{w}$, $\mathbf{w}(0) = \mathbf{w}_0$ is defined on a maximal interval $[0, \bar{t})$ that cannot be extended further to the right.*

The meaning of result (33) is that, depending on the initial conditions and on the eigenvalues of the covariance $\mathbf{\Phi}$, the learning rule (8) may diverge, also after a finite time-interval.

2.3 Stabilization of FMCA algorithm

In order to overcome the theoretical difficulties introduced by the previous observations, Oja presented an algorithm that is still based on the same objective function minimization by a gradient-descent method, but that had a slightly different structure [42].

Let us consider again the problem of minimizing the cost function:

$$C(\mathbf{w}) \stackrel{\mathrm{def}}{=} \frac{1}{2}E_{\mathbf{x}}[(\mathbf{w}^T\mathbf{x})^2|\mathbf{w}] + \frac{\lambda}{2}(\mathbf{w}^T\mathbf{w} - 1) \ , \tag{9}$$

with respect to the weight vector $\mathbf{w}$. Its gradient has the expression $\frac{\partial C}{\partial \mathbf{w}} = E_{\mathbf{x}}[y\mathbf{x}|\mathbf{w}] + \lambda\mathbf{w}$, thus the optimal multiplier may be found by vanishing $\mathbf{w}^T\frac{\partial C}{\partial \mathbf{w}}$, that is by solving:

$$\begin{cases} \mathbf{w}^T\frac{\partial C}{\partial \mathbf{w}} = E_{\mathbf{x}}[y^2|\mathbf{w}] + \lambda\mathbf{w}^T\mathbf{w} \ , \\ \mathbf{w}^T\mathbf{w} = 1 \ . \end{cases}$$

Now the main point is to recognize that, from an optimization point of view, the above system is *equivalent* to:

$$\begin{cases} \mathbf{w}^T \frac{\partial C}{\partial \mathbf{w}} = E_{\mathbf{x}}[y^2|\mathbf{w}] + \lambda - \bar{\sigma}(\mathbf{w}^T\mathbf{w} - 1) \ , \\ \mathbf{w}^T\mathbf{w} = 1 \ . \end{cases}$$

with $\bar{\sigma}$ being an arbitrary constant. This way, by computing the optimal multiplier we obtain the stabilized learning rule S-FMCA:

$$\begin{cases} \frac{d\mathbf{w}(t)}{dt} = -E_{\mathbf{x}}[y(t)\mathbf{x}(t) - y^2(t)\mathbf{w}(t)|\mathbf{w}] - \bar{\sigma}[\mathbf{w}^T(t)\mathbf{w}(t) - 1]\mathbf{w}(t) \ , \\ \mathbf{w}(0) = \mathbf{w}_0 \ . \end{cases} \tag{10}$$

It is possible to prove that the S-FMCA converges to the expected solution providing that the constant $\bar{\sigma}$ be properly chosen. The results about stabilization compactly recast in the following Theorem.

Theorem 3 *Let* $\mathbf{\Phi} \stackrel{\text{def}}{=} E_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T]$ *be the positive-definite covariance matrix of the random process* $\mathbf{x}(t)$ *in (10) with eigenpairs* $(\sigma_1, \mathbf{q}_1)$, $\ldots$, $(\sigma_p, \mathbf{q}_p)$*. Suppose eigenvalues are distinct and arranged in descending order, eigenvectors are normalized so that* $\mathbf{q}_k^T\mathbf{q}_k = 1$*, and* $\mathbf{w}_0^T\mathbf{q}_p \neq 0$*. If* $\bar{\sigma} > \sigma_1$ *then the state vector* $\mathbf{w}$ *of system (10) asymptotically converges towards* $+\mathbf{q}_p$ *or* $-\mathbf{q}_p$*.*

Proof. System (10) can be rewritten as:

$$\dot{\mathbf{w}} = -(\mathbf{\Phi} - \bar{\sigma}\mathbf{I})\mathbf{w} + \mathbf{w}^T(\mathbf{\Phi} - \bar{\sigma}\mathbf{I})\mathbf{w}\mathbf{w} \ . \tag{11}$$

Let us define:

$$\bar{\mathbf{\Phi}} \stackrel{\text{def}}{=} -(\mathbf{\Phi} - \bar{\sigma}\mathbf{I}) \ .$$

The eigenvalues of $\bar{\mathbf{\Phi}}$ are $\bar{\sigma} - \sigma_p > \bar{\sigma} - \sigma_{p-1} > \cdots > \bar{\sigma} - \sigma_1 > 0$, while its eigenvectors coincide to the eigenvectors of $\mathbf{\Phi}$. Thus Theorem 1 applies to system $\dot{\mathbf{w}} = \bar{\mathbf{\Phi}}\mathbf{w} + (\mathbf{w}^T\bar{\mathbf{\Phi}}\mathbf{w})\mathbf{w}$ allowing to conclude that $\mathbf{w}$ asymptotically converges to the last eigenvector $\mathbf{q}_p$ except for the sign. □

It is important to note that the Theorem just proven only gives a *sufficient* condition for the stability of S-FMCA, not a necessary one. It is also worth noticing that the introduced stabilization is what is called 'origin shift' in the context of numerical methods for matrix eigenvalue problems, which has been introduced here by the help of Lagrange multipliers and Kuhn-Tucker theory.

3 Discussion and numerical examples

In the following, some considerations are carried out on the implementation of FPCA and S-FMCA learning algorithms and on the relationships among the discussed adaptive principal/minor component analysis theory and some related standard signal processing techniques.

Also, computer simulation results are shown on synthetic signals to numerically assess the analytical results reported in the previous sections. Particularly, it is of interest to simulate the behavior of the principal modes both in FPCA and FMCA rules in order to numerically illustrate the consequences of Theorem 3.

3.1 Practical considerations on implementation

Both FPCA and FMCA algorithms are generally employed under a very realistic hypothesis, that is the second-order statistics of the involved signals are unknown, thus the covariance matrix $\mathbf{\Phi}$ is not available and its estimation would require both an adequate storage capacity and non-negligible computational efforts. Moreover, in most applications non-stationary signals should be dealt with, thus learning algorithms should be able to continually track covariance matrix eigenpairs.

In practical computer-based implementations, the discrete-time counterparts of the above learning equations are necessary. The simplest way for determining a discrete-time counterpart of learning equations described before is to employ the standard sampling method, consisting in determining a sufficiently narrow time-slice where the learning variables are almost stationary, say T, and replacing derivative $d\mathbf{w}/dt$ with $\Delta\mathbf{w}/T$, where $\Delta\mathbf{w} = \mathbf{w}((n+1)T) - \mathbf{w}(nT)$ and now $n \in \mathbb{Z}$ denotes the discrete-time index; hereafter discrete-time (sampled) versions of continuous time signal, e.g. $\mathbf{x}(nT)$, will be denotes as $\mathbf{x}[n]$. In this case we may resort to stochastic adaptation which finds its roots in sequential parameter estimation.

Sequential methods for parameter estimation rely on iterative algorithms to update the values of parameters as new data become available. These methods play an important role in signal processing and pattern recognition for three main reasons: 1) They do not require to store a complete data-set since each datum can be discarded once it has been used, making them very efficient when large data volumes are to be handled with; 2) They can be employed for online learning in real-time adaptive circuits; 3) In case of operation under non-stationary conditions, i.e. when the process that generates the data has slowly-varying statistical features, the parameters values can continuously adapt and can therefore track the behavior of the process.

From a more formal viewpoint, the invoked adaptive algorithms may be regarded as procedures for finding the roots of functions which are defined stochastically. To give an example, let us consider two scalar variables, u and w, which are correlated; the average of u for each w defines a function $g(w) \stackrel{\text{def}}{=} E_u[u|w]$. In the hypothesis that several observations of the variable u for a given value of w are available, we have a set of random values whose average value g, thought of as a function of w, is usually referred to as *regression function*. A general procedure for finding the roots $w^\star$ of such function was given by Robbins and Monro [45], which reads:

$$w[n+1] = w[n] + \mu_n u(w[n]) \ ;$$

under fair conditions on u, g and on the sequence of learning step-sizes μ_n, it can be proven that the sequence of estimates $w[n]$ converges to one of the roots $w^\star$ with probability 1. Such stochastic sequential approximation scheme was extended to the multidimensional case by Blumm [6]. Also, it is a common practice to take μ constant at a sufficiently small value which ensures good convergence in a reasonably short time.

The most exploited solution to the mentioned problems thus consists in invoking the discussed discrete-time stochastic versions of FPCA and FMCA rules. Within this framework, learning rules (5) and (10) recast, respectively, into:

$$\Delta \mathbf{w} = \mu(y\mathbf{x} - y^2\mathbf{w}) \ , \ \mathbf{w}(0) = \mathbf{w}_0 \ , \tag{12}$$

$$\Delta \mathbf{w} = -\mu(y\mathbf{x} - y^2\mathbf{w}) - \mu\bar{\sigma}(\mathbf{w}^T\mathbf{w} - 1)\mathbf{w} \ , \ \mathbf{w}(0) = \mathbf{w}_0 \ , \tag{13}$$

which represent the discrete-time stochastic counterpart of FPCA and S-FMCA rules. These equations may be easily implemented on a digital computer and exhibit minimal storage/computational requirements.

About stability Theorem 3, we know that the choice $\bar{\sigma} > \sigma_1$ ensures the convergence of S-FMCA rule by the 'origin shift' property. Some authors claim that the origin shift is not so profitable because it requires the prior knowledge of the largest eigenvalue. An interesting observation concerning this condition is that the a-priori knowledge of eigenvalue σ_1 *is not actually required*; in fact it is possible to choose $\bar{\sigma} = \mathrm{tr}(\mathbf{\Phi}) = \sum_i \sigma_i > \sigma_1$. It is worth noting that $\mathrm{tr}(\mathbf{\Phi}) = E_{\mathbf{x}}[\mathbf{x}^T\mathbf{x}]$ just represents the power of input process which may be easily measurable; alternatively, as it is supposed to be a bounded quantity, an upper bound for $\mathrm{tr}(\mathbf{\Phi})$ may be used instead.

3.2 A numerical example on synthetic signals

Consider an input random process $\mathbf{x} \in \mathbb{R}^4$ whose covariance matrix $\mathbf{\Phi}$ is as in [10]. The input signal is obtained by $\mathbf{x}[n] = \mathbf{A}\mathbf{\Sigma}^{1/2}\mathbf{g}[n]$, with $\mathbf{g}[n]$ being white

Gaussian (normal) noise and $\mathbf{A\Sigma A}^T = \mathbf{\Phi}$. Also, in our experiments we had the entries of $\mathbf{w}_0$ randomly (uniformly) picked in $[-1, +1]$ and investigated the behavior of principal modes $\theta_i[n] = \mathbf{w}^T[n]\mathbf{a}_i$; here $\mathbf{a}_i$ stands for the i^{th} column of $\mathbf{A}$ and we supposed the eigenvalues on the diagonal of $\mathbf{\Sigma}$ be arranged in descending order.

Running the learning rule (12) with $\mu = 0.005$, that allows to extract the first principal component from the input stream, one expects that the first principal mode θ_1 tends to $+1$ or -1, while second, third and fourth principal modes tend to zero. These results are confirmed by the results shown in the Figure 2, which depicts the behavior of $|\theta_i|$ versus learning iterations; as convergence may depend on initial conditions, the shown curves are averaged over 20 independent trials.

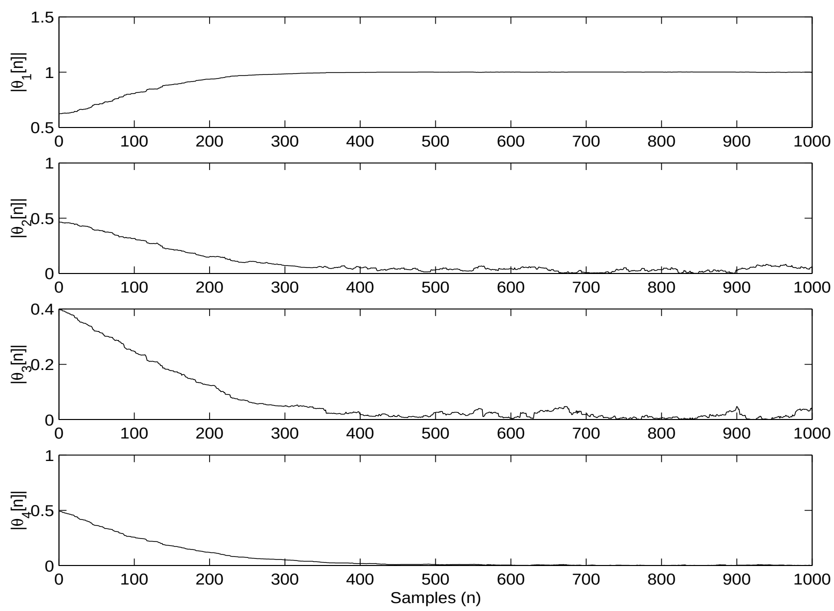

Figure 2: Absolute value of principal modes for FPCA.

Furthermore, we tried to run the learning rule (13) on the same data set in order to extract the first minor component. We tried first with $\bar{\sigma} = 0$, that means using the non-stabilized first minor component analyzer FMCA: Simulations show that the rule diverges quickly regardless of the learning step-size. Then, with $\mu = 0.05$ we tried to use the sufficient condition provided by Theorem 3, which drove us to the choices $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.0$, $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.5$, $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 2.0$ and $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 2.5$. Simulation results are shown in Figures 3, 4, 5 and 6, respectively; the shown curves are averaged over 20 independent trials. As expected, the first three principal modes converge to zero, while fourth mode module approaches 1.

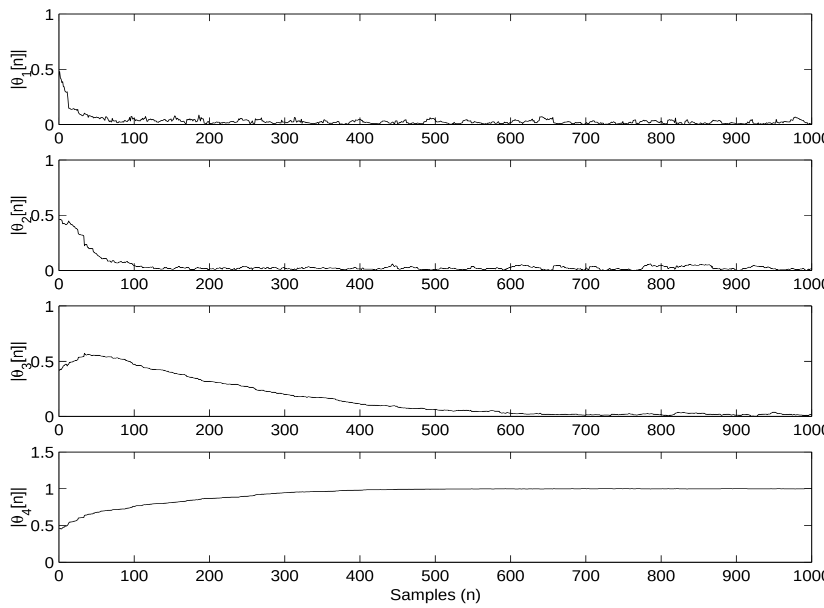

Figure 3: Principal mode modules for FMCA, for $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.0$.

3.3 Relationships with standard signal processing techniques

The discussed algorithms are developed under the hypothesis that the statistics of the signals is not known in advance (nor is it covenient to estimate), that is the covariance matrix of the signal $\mathbf{x}(t)$ is not accessible.

When the covariance matrix $\mathbf{\Phi}$ is known and the problem is to estimate associated eigenspaces, the principal component convergence can be compared to standard linear algebra packages, like the Cholesky and the SVD decomposition. Indeed, with the covariance matrix we may invoke the Gauss-Markov Theorem to get estimates of all kinds of subspaces [3]; but implicit in this supposition is that the sampled random process $\mathbf{x}[n]$ is obtained as $\mathbf{x}[n] = \mathbf{A}\mathbf{g}[n]$, where $\mathbf{g}[n]$ is white noise and $\mathbf{\Phi} = \mathbf{A}\mathbf{A}^T$, or has the form:

$$\mathbf{x}[n] = \sum_{m=0}^{+\infty} \mathbf{A}_m \mathbf{g}[n-m] \ ;$$

with the filter $\mathbf{A}_m$ being summable; in this case the comparison should be made between the principal-component technique and the well-established system identification methods that have been developed during the last 50 years.

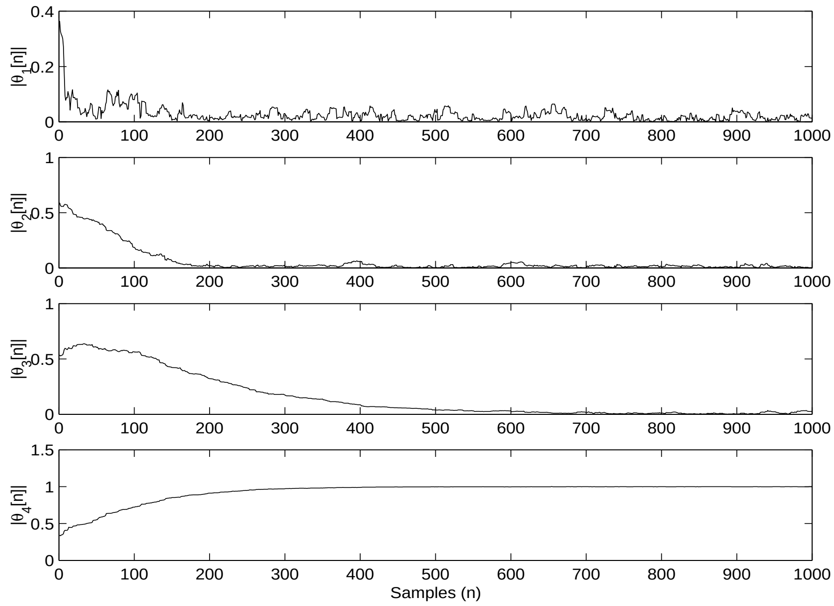

Figure 4: Principal mode modules for FMCA, for $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.5$.

In case of non-stationary signals, which might be represented by:

$$\mathbf{x}[n] = \sum_{m=0}^{+\infty} \mathbf{A}_m[n]\mathbf{g}[n-m] \ ,$$

and of known statistics, a comparison could be made against standard adaptive filtering techniques.

Conversely, let us consider data reduction techniques which aim at providing an efficient representation of the data; we may focus on the procedure consisting in mapping the higher dimensional data-space into a lower dimensional representation space by means of a linear transformation, as in the Karhunen-Loéve Transform (KLT). The classical approach for evaluating the KLT requires the computation of the input data covariance matrix and then the application of a numerical procedure to extract the eigenvalues and the corresponding eigenvectors; compression is obtained by the use of the only eigenvectors associated with the most significant eigenvalues as a new basis. When large data sets are handled, this approach is not practicable because the dimensions of the covariance matrix become too large to be manipulated, and the whole set of eigenvectors has to be evaluated even though only a little amount of them are truly used.

This short discussion partially explains the success of adaptive principal component analysis techniques which have seen a great theoretical effort to be made more and more powerful and efficient, until the recent development of

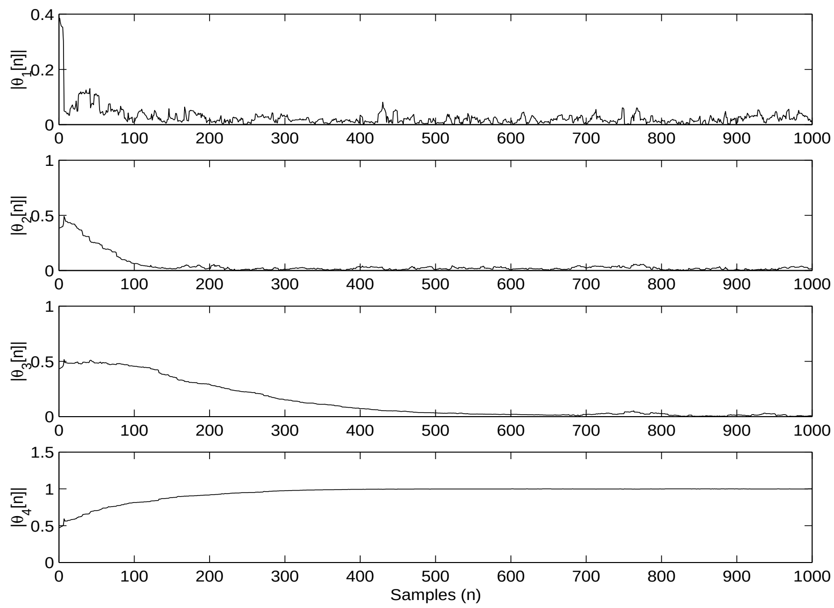

Figure 5: Principal mode modules for FMCA, for $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 2.0$.

very fast and reliable neural algorithms (see, for instance, [9]); some of them are based on single-unit networks which just exploits the concept of deflation to extract more than one principal/minor component [12].

4 Robust-Constrained Beamforming by FMCA

We consider a minor component analysis approach to robust beamforming with control of array beampattern by constrained adaptation. The proposed approach arises as a case of variance minimization for a linear neural unit. In details, a constrained beamformer power optimization principle may be employed, which allows to improve the performances of simpler beamforming algorithms by emphasizing white noise sensitivity control and prior knowledge about the disturbances. The present approach stems from the algorithm introduced in the preliminary report [22].

4.1 Existing contributions to neural adaptive antenna array signal processing

Antenna array signal processing mainly consists of direction-of-arrival (DoA) estimation and antenna beamforming. Some systems need only DoA estimation to detect the signals, such as radar or sonar, while others need beamforming to

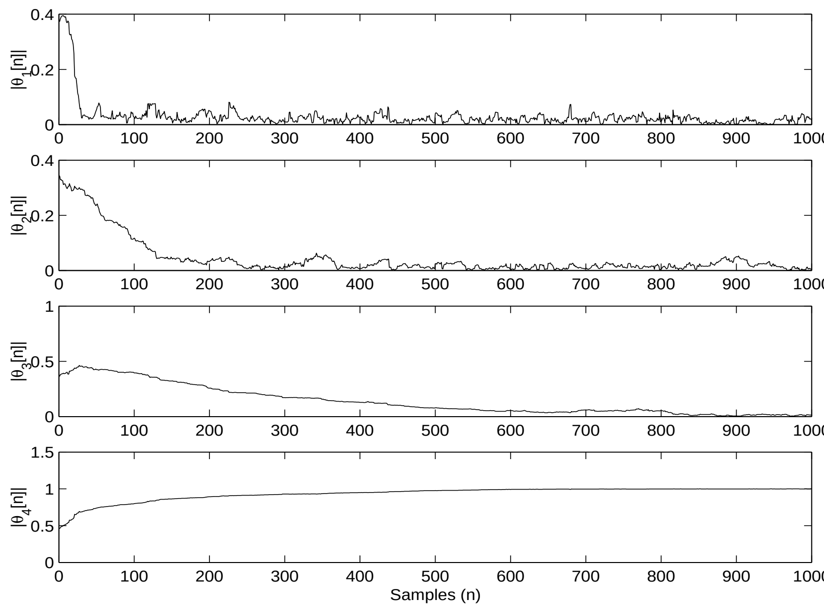

Figure 6: Principal mode modules for FMCA, for $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 2.5$.

acquire the relevant signals such as in mobile communications.

Beamforming methods may be classified into two categories: Beamforming based on DoA estimated by a calibrated array and beamforming based on a known training signal transmitted by the user. There are also some blind beamforming methods, which do not require the knowledge of the DoA nor of a training sequence [].

Generally, the existing methods cannot meet the requirements of real time and multi-source tracking requirements. Neural-network-based methods are typically adaptive method which proved to provide powerful general-purpose algorithms. The neural models for the two classes of problems may be described as follows: An antenna array works as a non-linear mapping from source signals to array measurements, while the two classes of problems are both inversion problems, trying to solve for the source signals on the basis of the array measurements. An excellent review of neural-network-based methods for direction-of-arrival estimation and beamforming has been presented recently by Du *et al.* in [18].

The DoA problem aims to get the DoA of signals from the measurement of the array output, while beamforming technique tries to recover the original signal of the desired source. For an antenna array system, a neural network is first trained, which then performs the DoA estimation or beamforming. Here we recall the principal neural methods known in the literature involving principal component analysis and its extension known as independent component analysis (ICA).

Two algorithms are at present known that exploit the concept of cyclostationarity. A cross-correlation asymmetric PCA network-based beamformer which makes use of the cyclostationary property of signals to perform blind beamforming has been proposed in [31]. It is formulated as an SVD problem of the correlation matrix of the array data and its time-frequency translated version. In [19], a fast, sub-optimal blind cyclostationary beamforming algorithm inspired from the cross-correlation asymmetric PCA network of Kung and Diamanatars (for a review see [13]) was been proposed.

Most of the antenna array signal processing methods are based on the subspace concept and require the eigen-decomposition of the input correlation matrix. With the aim to improve the well-known MUSIC algorithm, many efforts have been made to compute the noise space as quickly as possible to meet the real-time requirement. An MCA or PCA algorithm has been proposed in [4] to extract the noise or signal subspace, respectively. By transforming the eigenvector problem in complex form into that in real form, one can perform the extraction of the noise (or signal) subspace using the anti-Hebbian (or Oja) algorithrn (see e.g. citecosta), respectively. This produces an iterative procedure for real-time DoA estimation and tracking. Such algorithm has demonstrated the capability of tracking two time-changing sources by simulation. In [8], the unitary decomposition neural network (UNIDANN) was described. The UNIDANN can perform the unitary eigen-decomposition of a Hermitian positive definite synaptic weight-matrix, in fact the neural output will converge to the principal eigenvectors of the synaptic weight matrix. Due to the introduction of an optimal time-varying weighing in the recursive equation (and its underlying analog circuit structure), the UNIDANN exhibits a fast rate of convergence and excellent numerical stability.

As mentioned, higher-order statistical techniques than principal/minor component analysis may be employed in blind beamforming, such as blind source separation and independent component analysis.

In [7], the Authors consider an application of blind identification to beamforming through the use of estimates of directional vectors rather than their hypothesized value. Blind identification allows array signal prcessing without knowing the arrray manifold and thus the beamforming is made robust with respect to array deformations, distortion of the wave front and pointing errors, so that neither array calibration nor physical modeling are necessary. A surprising result first emerging from this contribution is that 'blind beamformers' may outperform 'informed beamformers' even when the array is perfectly known to the informed beamformer. The key assumption is the statistical independence of the signal sources, which is exploited by the help of fourth-order statistics.

In [38], the Authors suggested to combine a symmetrically balanced beamforming array with the Herault-Jutten neural network for separating out independent broadband sound sources and their multipath delays. The advantages of the proposed method are that no penalty occurs for long impulse responses caused by long delays and no training signals needed for separation.

In the recent contribution [27], a possible approach to electromagnetic broadband source localization using a hybrid blind-separation/inversion algorithm was proposed. The total electrical field versus time, emitted by the working antennas located at different and unknown geographical positions, is used, via a suitable blind signal processing technique based on neural networks, to reconstruct each separate contribute. When the emitted electric signals have been separated for each emitting antenna, the unknown locations of the antennas are determined with a numerical inversion technique.

4.2 Adaptive beamforming and constrained optimization

A on-line beamformer may be realized by a linear neural unit described by a *complex* weight-vector $\mathbf{w} \in \mathbb{C}^p$, where p is the number of sensors, and the input-output relationship $y[n,\omega] = \mathbf{w}^H[n,\omega]\mathbf{x}[n,\omega]$, where '^{H}' denotes Hermitian conjugation. The input $\mathbf{x}$ usually contains the discrete Fourier transform of the sampled signals coming from the sensors, whereby its complex nature; the quantity ω denotes the frequency bin under consideration. It is very important to specify the frequency bin under consideration, because the array beampattern depends on ω or, equivalently, a signal 'sees' a different beampattern depending on its frequency; this means that an array may cause a distortion of a non-harmonic signal because each of its harmonic components *theoretically impinges a different array.* This problem is negligible for so-called narrow-band signals, which have sufficiently narrow spectra around a central value ω_c that ω may be assumed equal to. Also, the geometry of the sensors array, i.e. the location of sensors and their distance one from another, influences the array beampattern. For instance, a classical configuration is the *end-fire*, which consists of a number of sensors placed in a linear array at regular distances one from another [14]. The directional response of each elementary sensor is another factor influencing the behavior of an array. Elementary sensors may be omni-directional (panoramic) or directional, i.e. the sensor is more sensitive in some directions; sometimes, for the sake of easy treatment, panoramic sensor hypothesis is considered, which allows for a better control of the directionality of the whole array.

By restricting to planar geometry, the array configuration is expressed by a steering vector $\mathbf{d}(\theta)$ defined as the vector of phase delays to align the array outputs for a plane wave coming from direction θ. Also, it is useful to decompose the covariance matrix $\mathbf{\Phi}$ of the array input signal $\mathbf{x}$ (also know as spectral covariance matrix) into signal and noise components as follows:

$$\mathbf{\Phi} = \sigma_s^2 \mathbf{d}\mathbf{d}^H + \sigma_n^2 \mathbf{N} \ . \tag{14}$$

The noise cross-spectral matrix $\mathbf{N}$ is normalized to have its trace equal to the number of sensors p so that σ_s^2/σ_n^2 is the signal-to-noise spectral ratio averaged over the p sensors. The array gain $G(\theta)$ represents the improvement of signal-to-noise ratio along direction θ, owing to beamforming, that is:

$$G(\theta) \stackrel{\text{def}}{=} \frac{|\mathbf{w}^H \mathbf{d}(\theta)|^2}{\mathbf{w}^H \mathbf{N} \mathbf{w}} \ . \tag{15}$$

The profile of array response at different values of θ is usually referred to as *array beampattern*.

When the noise is spatially white, i.e. $\mathbf{N} = \mathbf{I}_p$, the array gain becomes what is called white noise gain $G_w(\theta) \stackrel{\text{def}}{=} \frac{|\mathbf{w}^H \mathbf{d}(\theta)|^2}{\mathbf{w}^H \mathbf{w}} \leq p$ [14]. Furthermore, it is defined the sensitivity $S(\theta)$ of array gain to signal mismatching by considering the signal $\mathbf{x}$ perturbed by small random zero-mean errors; when the errors are uncorrelated from sensor to sensor, the sensitivity, $S_w(\theta)$, proves to equal $G_w^{-1}(\theta)$ [14]. Prescribing a bound for the sensitivity of the array usually means constraining the white noise gain in the looking direction as $G_w(\theta_s) = \delta^2$, where the value δ^2 must be chosen less or equal to p for the constraint to be consistent. The white noise gain is retained as a useful and convenient measure of robustness: Large values of δ correspond to strong robustness, while small values of δ do not provide robust design.

By denoting with θ_s the direction of arrival of the primary source and with $\mathbf{d}_s \stackrel{\text{def}}{=} \mathbf{d}(\theta_s)$ the corresponding steering vector, a constraint which is usually considered is the *unit boresight response*, that writes $\mathbf{w}^H \mathbf{d}_s = 1$ and ensures no attenuation and zero phase-shift in the direction of arrival of primary source.

A way to train the beamforming neuron is to design a learning rule as a system to solve the following optimization problem [14]:

$$\min_{\mathbf{w} \in \mathbb{C}^p} E_{\mathbf{x}}[|\mathbf{w}^H \mathbf{x}|^2 | \mathbf{w}] \ , \mathbf{w}^H \mathbf{w} = \delta^{-2} \ . \tag{16}$$

It is worth noticing that the optimization problem above is a power minimization one; thus we may use the S-FMCA learning rule extended to the complex domain in order to train the beamforming neuron, and the counterpart of the Theorem 3 to ensure its convergence. In particular, the cost function to be minimized by a gradient steepest descent learning rule is now:

$$C_{\mathrm{BF}}(\mathbf{w}) = 0.5E_{\mathbf{x}}[|y|^2|\mathbf{w}] + 0.5\lambda(\mathbf{w}^H\mathbf{w} - \delta^{-2}) \ . \qquad (17)$$

On the basis of cost function (17), the convergence Theorem below holds (the proof is omitted because it is a straightforward extension of the proof of Theorem 3).

Theorem 4 *Let* $\mathbf{\Phi} \stackrel{\mathrm{def}}{=} E_{\mathbf{x}}[\mathbf{x}\mathbf{x}^H]$ *be the spectral covariance matrix of the random process* $\mathbf{x}(t)$ *in (17) with eigenpairs* $(\sigma_1, \mathbf{q}_1)$, $\ldots$, $(\sigma_p, \mathbf{q}_p)$*. Suppose eigenvalues are distinct and arranged in descending order, and eigenvectors are normalized so that* $\mathbf{q}_k^H\mathbf{q}_k = 1$*. If* $\bar{\sigma} > \delta^{-2}\sigma_1$ *then the state vector* $\mathbf{w}(t)$ *of gradient steepest minimization system of cost function (17), with initial state* $\mathbf{w}_0^H\mathbf{q}_p \neq 0$*, asymptotically converges towards* $\mathbf{q}_p$ *up to phase shift.*

As an improvement to above theory, Cox *et al.* report and discuss the addition of some constraints allowing for better controlling white noise sensitivity and for exploiting the prior knowledge about the geometry of the array and about the disturbances [14]. In details, an enhanced way to train the beamforming neuron is to teach it to solve the following optimization problem [14]:

$$\min_{\mathbf{w}} \mathbf{w}^H\mathbf{\Phi}\mathbf{w} \ , \ \mathbf{w}^H\mathbf{w} = \delta^{-2} \ , \ \mathbf{K}^H\mathbf{w} = \mathbf{b} \ , \qquad (18)$$

where $\mathbf{\Phi}$ is again the spectral covariance matrix of the input, δ is again the constant that limits the white noise sensitivity, and $\mathbf{K}^H\mathbf{w} = \mathbf{b}$ is a set of linear constraints used to improve the performances of the beamformer by accounting for prior knowledge; in the context of adaptive filtering, it has been exploited by Frost [14]. In particular, $\mathbf{K} \in \mathbb{C}^{p\times k}$ has $k < p$ linearly independent columns, and $\mathbf{b} \in \mathbb{C}^k$. One column of $\mathbf{K}$ is usually $\mathbf{d}_s$, and the corresponding entry of $\mathbf{b}$ is 1, in order to include in (18) the unit boresight response constraint.

5 Experimental results

Here we refer to the triangular geometry depicted in Figure 7, thought to for teleconference applications, enclosing 3 microphones (MIC) and a loudspeaker in the center of the triangle (LS). For this configuration the steering vector finds to be:

$$\mathbf{d}^H(\theta) = \left[e^{\frac{-j\pi r}{\sqrt{3}}(\sqrt{3}\cos(\theta)+\sin(\theta))} \ e^{\frac{j\pi r}{\sqrt{3}}(\sqrt{3}\cos(\theta)-\sin(\theta))} \ e^{\frac{2j\pi r}{\sqrt{3}}\sin(\theta)} \right] \ , \qquad (19)$$

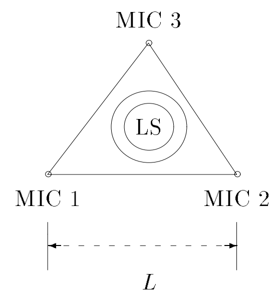

Figure 7: Microphone array geometry (LS = Loudspeaker, MIC = Microphone).

where $r \stackrel{\text{def}}{=} \frac{\omega L}{2\pi c}$. Here ω denotes again the angular frequency corresponding to the considered frequency bin, L denotes the distance among the microphones, and c is the sound speed. The reference system originates in the center of the triangle and θ is the angle formed by an incoming beam with the x-axis, as shown in Figure 8.

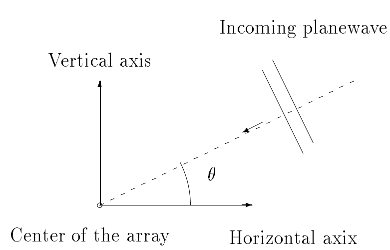

Figure 8: Reference system and θ angle.

In order to adaptively enforce the constraints to be met, let us define the quantities:

$$\bar{\mathbf{P}} \stackrel{\text{def}}{=} \mathbf{I}_p - \mathbf{K}(\mathbf{K}^H\mathbf{K})^{-1}\mathbf{K}^H \ , \bar{\mathbf{w}} \stackrel{\text{def}}{=} \mathbf{K}(\mathbf{K}^H\mathbf{K})^{-1}\mathbf{b} \ .$$

In [14] it is proposed an algorithm for updating the weight vector so that both constraints in (18) are fulfilled while minimizing the spectral power of the neuron's output $C(\mathbf{w}) \stackrel{\text{def}}{=} E_{\mathbf{x}}[|y|^2|\mathbf{w}]$. In practice, this algorithm can be expressed as:

$$\mathbf{w}(t+1) = \bar{\mathbf{w}} + \bar{\mathbf{P}}\mathbf{w}(t) - \mu\bar{\mathbf{P}}\frac{\partial C(\mathbf{w})}{\partial \mathbf{w}(t)} \ , \tag{20}$$

plus a mechanism for optimizing on the sphere $\mathbf{w}^H\mathbf{w} = \delta^{-2}$. Here we use instead the S-FMCA cost function C_{BF} leading to the gradient:

$$\frac{\partial C_{\mathrm{BF}}(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{\Phi}\mathbf{w} - \delta^2(\mathbf{w}^H\mathbf{\Phi}\mathbf{w})\mathbf{w} + \bar{\sigma}(\mathbf{w}^H\mathbf{w} - \delta^{-2})\mathbf{w} \ .$$

It allows to satisfy both constraints without additional mechanisms.

The stochastic counterpart of above gradient reads:

$$\frac{\partial C_{\mathrm{BF}}(\mathbf{w})}{\partial \mathbf{w}} \approx \mathbf{x}y^\star - \delta^2|y|^2\mathbf{w} + \bar{\sigma}(\mathbf{w}^H\mathbf{w} - \delta^{-2})\mathbf{w} \ , \tag{21}$$

as used for computer-based implementation; in the above formula '$\star$' denotes complex conjugation.

The array number r in equation (19) takes into account both array size and considered frequency bin, thus it represents an important feature of the system of sensors and deserves a short discussion closely related to *aliasing* phenomenon. In fact, from basic principles of wave physics we find that $r = \frac{L}{\lambda}$, where λ is the wavelength corresponding to the frequency-slice that the array is accorded to; by supposing that an incoming plane wave can impinge the array from whatever direction, we see that the plane wave on its path encounters two sensors (say, for instance, MIC 1 and MIC 2), thus is spatially sampled in two points. In order to avoid ambiguities, according to Shannon-Hartley sampling theorem, we know that the sampling frequency must be at least twice the maximum signal frequency. In this case the sampling frequency relates to $\frac{1}{L}$ and the wave frequency is $\frac{1}{\lambda}$, thus the following important relationship should hold:

$$r = \frac{L}{\lambda} \leq 0.5 \ ;$$

in the context of array processing, the violation of this condition causes so-called *secondary lobes* to appear on the array beampattern.

It might be interesting to recall the *delay-and-sum* (DaS) beamformer in order to numerically verify the above qualitative observations. The DaS is the simplest beamformer known in the signal processing literature and is non-adaptively designed as $\mathbf{w} = \mathbf{d}_s/p$; it bases on a very simple idea: Under the hypothesis of plane-wave propagation of the primary signal and of uncorrelated disturbances coming from everywhere, the components of the primary signals interfere on the array and, upon exact phase alignment provided by the beamformer, outputs the array with unaltered power; conversely, the components of the noises interfere in a destructive way as they can come from everywhere and cannot be aligned.

The DaS array beampattern corresponding to six values of r for the considered triangular geometry is depicted in Figure 9, under the hypothesis that the primary source signal is known to impinge the array from $\theta_s = \pi/2$.

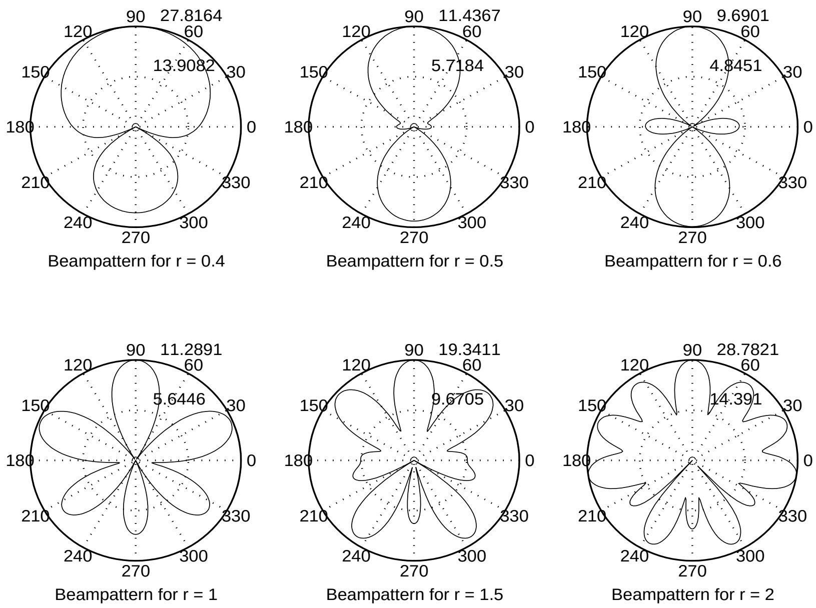

Figure 9: DaS beamformer beampattern $G(\theta)$ for six values of r (arbitrary scales in polar coordinate system).

Clearly the DaS beamformer automatically meets the unit boresight response condition, but does not embody any additional restriction nor allows for sensitivity control.

In the following we present some experimental results with the adaptive beamforming neuron performed on the basis of the triangular array.

5.1 Experiment on adaptive beamforming 1

In this experiment, suggested in [14], we test the behavior of the beamforming neuron in presence of weak source signal and strong white noise only (the central loudspeaker of the teleconferencing system is supposed to be turned-off and no strong directional disturbances are considered); for this experiment we chose, according to Cox *et al.*, $\sigma_n^2 = 0$ dB and $\sigma_s^2 = -10$ dB. As unit-boresight response is the only constraint, we used $\mathbf{K} = \mathbf{d}_s$ and $\mathbf{b} = 1$; also, we used the triangular beamformer with $r = 0.4$. The source signal is known to come from direction $\theta_s = \pi/2$ and we used the parameters $\mu = 0.0005$ and $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.5$.

The first row from the top of Figure 10 shows the signal enhancement for $\delta = 1.5$ as well as array beampattern after learning, while the second row refers to $\delta = 1$. It can be noted that the beamformer enhances the level of primary source signal against uncorrelated noise as long as robust design is

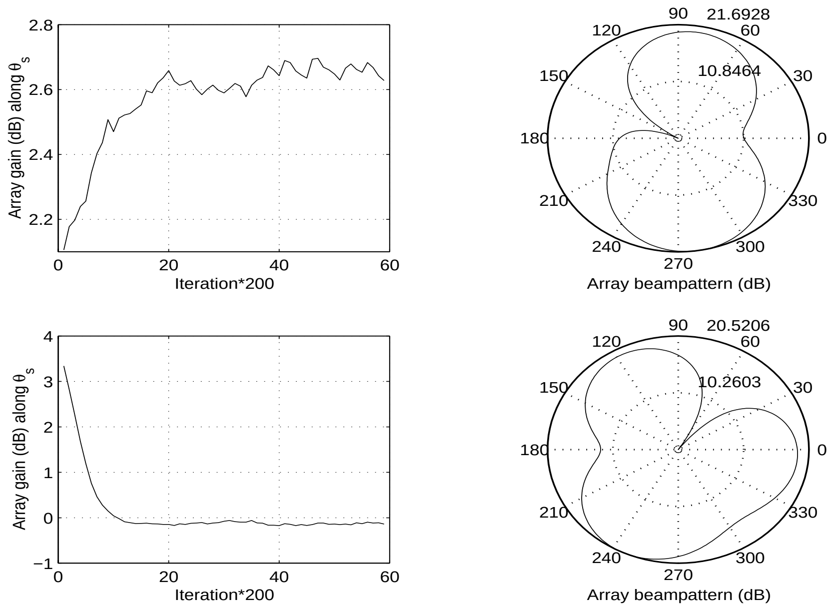

Figure 10: Experiment 1: Beamformer's performances. (Top row: $\delta = 1.5$. Bottom row: $\delta = 1$.)

enforced by $\delta = 1.5$, while the gain is unsatisfactory when weaker robustness is enforced. About array beampattern, it is interesting to note that, in absence of strong interference, the sidelobes are not critical to optimization and assume uncontrolled shape, as well as the 'null' position.

5.2 Experiment on adaptive beamforming 2

In this experiment, also suggested in [14], we test the behavior of the beamforming neuron with a strong primary source and white noise of comparable power, and in presence of array imperfections; for this experiment $\sigma_n^2 = 0$ dB, $\sigma_s^2 = 0$ dB. Again $\mathbf{K} = \mathbf{d}_s$ and $\mathbf{b} = 1$, while $r = 0.4$. Array imperfection is simulated by adding Gaussian uncorrelated errors to steering vector when generating the observed signal; it simulates the misalignment of sensors due, for instance, to mechanical shoves or imperfect construction; as the imperfections are unknown and unexpected, in the algorithm the 'nominal' steering vector is still used [14]. The source signal is known to come from direction $\theta_s = 0$ and we used the parameter $\mu = 0.0002$ and $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 10$.

The first row from top of Figure 11 shows the signal enhancement for $\delta = 1.5$ as well as array beampattern after learning and the second row refers to $\delta = 0.5$. In absence of strong white noise sensitivity control, the array beampattern looks

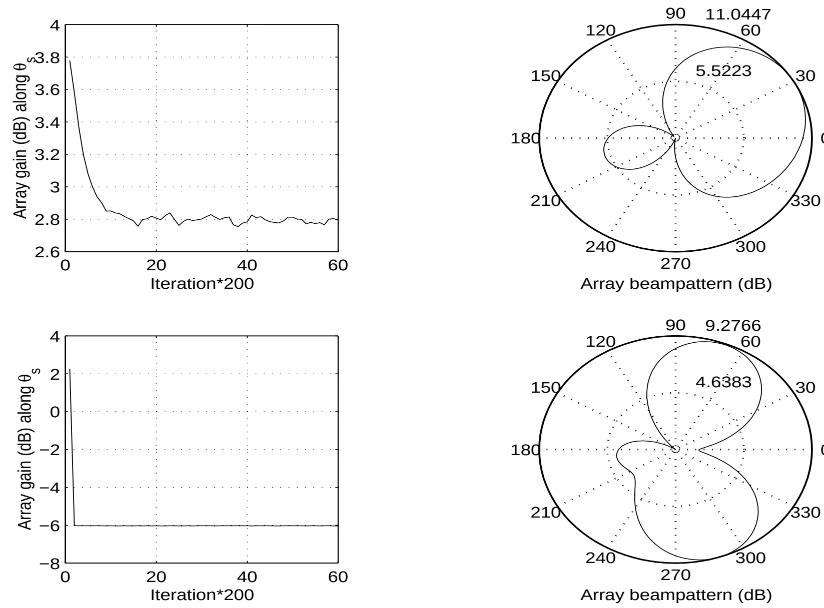

Figure 11: Experiment 2: Beamformer's performances. (Top row: $\delta = 1.5$. Bottom row: $\delta = 0.5$.)

unsatisfactory as, due to array misalignment, it happens that large portions of space interested by noise are enhanced; conversely, with better sensitivity control we can see a strong main-lobe around the direction of arrival of the primary signal and significant attenuation in the other zones of the space. Note that, in this case, the beamformer performs better with respect to the case of the Experiment 1 because of the higher signal-to-noise ratio.

5.3 Experiment on adaptive beamforming 3

In the present formulation of constrained beamforming, we use the pair $(\mathbf{K}, \mathbf{b})$ to achieve two results: Unit boresight response and zero-interference between the microphones MIC 1, 2, 3 and the central loudspeaker, that prevents unpleasant echoes and the well-known Larsen effect. The second target may be attained by imposing the constraint $w_1+w_2+w_3 = 0$: In this way the sound-waves traveling from the central loudspeaker to the lateral microphones through equilength paths cancel exactly at the output of the beamforming neuron. In this case we thus have $k = 2$ constraints on a $p = 3$-elements array. The constraint-pair in this case has thus the form $\mathbf{K} = [\mathbf{d}_s\ \mathbf{1}_3]$ and $\mathbf{b} = [1\ 0]^T$, where $\mathbf{1}_3 \stackrel{\text{def}}{=} [1\ 1\ 1]^T$.

Also, in the following experiments, the primary speech source signal is known to impinge the array from $\theta_s = 0$, and a directional disturbance impinges the array from $\theta_d = \pi/8$; also, spatially uncorrelated environmental noise is present. It is important to note the the information about the direction-of-arrival of the primary signal is exploited to design the spatial neural filter while the other information aren't.

Figure 12 shows the signal enhancement, noise rejection and white noise gain for $\delta = 1$, as well as the array beampattern after learning; Figure 13 refers to the same problem with a lower expected white noise gain, namely $\delta = 0.5$. In both experiments we took $\mu = 0.0005$ and $\bar{\sigma}/\mathrm{tr}(\mathbf{\Phi}) = 1.5$. In both cases the learning

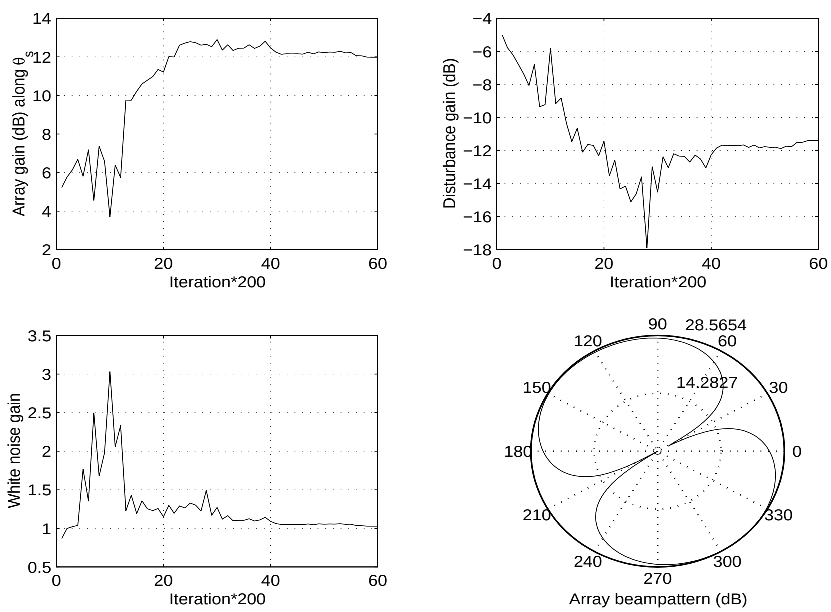

Figure 12: Experiment 3: Beamformer's performances for $\delta = 1$.

algorihm places the *null* of the array beampattern in correspondence of θ_d in order for the beamformer to suppress the strongest disturbance, and also shapes the beampattern in order to mitigate the environmental noise. However, again a stronger white noise sensitivity control allows for better performances; also, in these simulations algorithm has provided $\mathbf{d}_s^H \mathbf{w} - 1 \cong 0$ and $w_1 + w_2 + w_3 \cong 0$, as expected.

For these experiments it is interesting to have a closer inspection of the behavior of the beamformer by observing the primary speech signal enhancement as well as disturbance signals rejection. Figures 14 and 16 show the amplitude spectra of the four component of beamformer output in the frequency domain (namely, the contribution due exclusively to the primary source compared to the magnitude of the contributions due to the directional disturbance, the environmental white noise and the noise coming from the central loudspeaker) for the

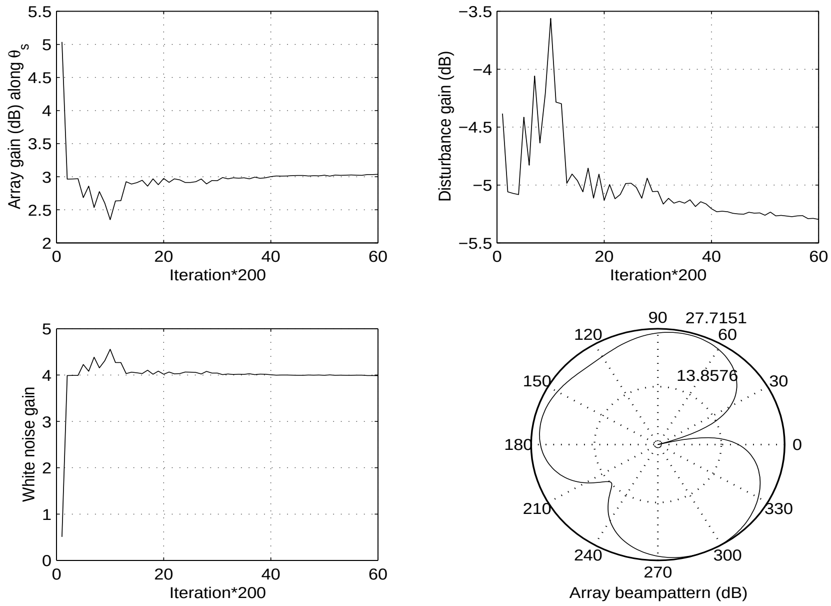

Figure 13: Experiment 3: Beamformer's performances for $\delta = 0.5$.

two values of the white noise sensitivity parameter. They show the directional disturbance rejection due to the 'null' in the array beampattern blindly placed by the FMCA algorithm as well as the attenuation of the white noise and the complete rejection of the loudspeaker signal. Figures 15 and 17 show instead the beamformer output both in the frequency and domain in the looking direction compared to the time/frequency representation of the true primary speech source.

6 Conclusion

The aim of this paper was to illustrate some results about first principal/minor component analysis by one-unit neural systems, and some stability and convergence properties of them; the reported material comes from the experience acquired by the author during the last five years of research in the PCA field.

We have recast formal results from the scientific literature stating that a version of Oja's neural MCA rule looks unstable, and we have reviewed a stabilization theory for it. Particularly, we have focused our attention on the mathematical structure and properties of the cited neural learning rules in order to gain knowledge on the problems and common solutions arising when dealing with PCA-like adaptive (neural) systems.

Neural adaptive first minor component extraction has been adapted to ro-

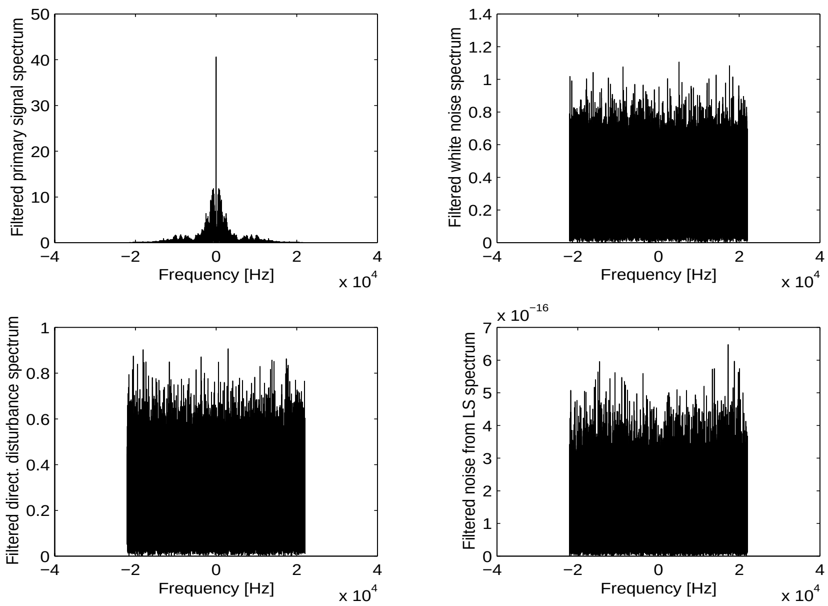

Figure 14: Experiment 3: Spectral contributions to beamformer output in the case $\delta = 1$.

bust constrained beamforming applied to a triangular microphone-array. Experimental results have illustrated the possible use of FMCA theory for designing an effective spatial-filtering neural structure endowed with an adaptive rule that embodies white-noise-gain control and zero-interarray-interference constraints.

Acknowledgments

I would like to gratefully thank Prof. F. Piazza (University of Ancona, Italy) for the interesting and fruitful discussions on adaptive beamforming during my Ph.D. studentship, Dr K.-L. Du (Concordia University, Canada) for sharing bibliographical information and computer codes on adaptive beamforming and the anonymous reviewers whose detailed and careful comments greatly helped to improve the quality of the presentation of the initial manuscript.

References

[1] H.M. Abbas and M.M. Fahmy, *Neural Model for Karhunen-Loéve transform with application to adaptive image compression*, IEE Proc. I, Communications, Speech and Vision, Vol. 140, No. 2, pp. 135 – 143, Apr. 1994

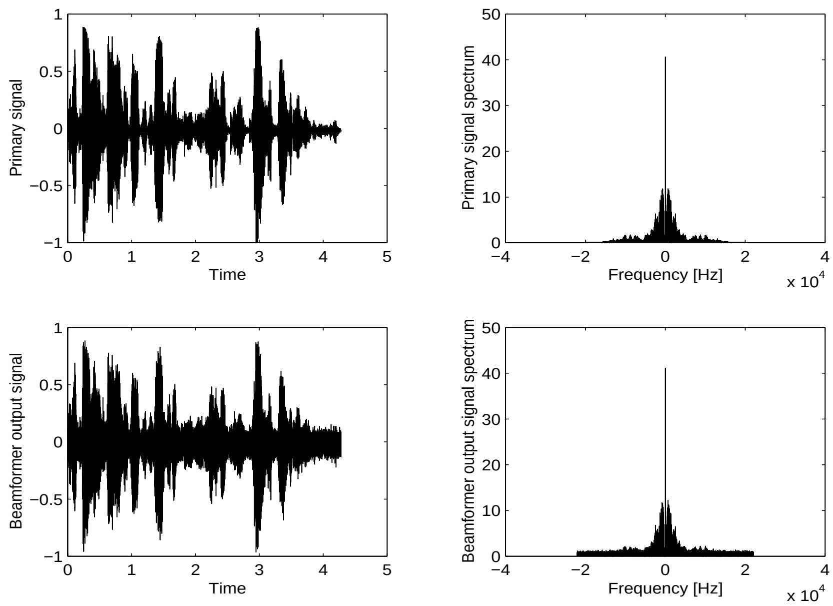

Figure 15: Experiment 3: Time/spectral beamformer output compared to time/spectral primary source in the case $\delta = 1$.

[2] S.-I. Amari, *Neural theory of association and concept formation*, Biological Cybernetics, Vol. 26, pp. 175 – 185, 1977

[3] Ph. Besse and J.O. Ramsay, *Principal components analysis of sampled functions*, Psychometrika, Vol. 51, No. 2, pp. 285 – 311, 1986

[4] L. Badidi and L. Radouane, *A neural network approach for DoA estimation and tracking*, Proc. of the 10th IEEE Workshop os Statistical Signal and Array Processing, pp. 434 – 438, August 2000

[5] S. Bannour and M.R. Azimi-Sadjadi, *Principal component extraction using recursive least squares learning*, IEEE Trans. on Neural Networks, Vol.6, No.2, pp. 457 – 469, March 1995

[6] R. Blumm, *Multidimensional stochastic approximation methods*, Annals of Mathematical Statistics, Vol. 25, pp. 737 – 744, 1954

[7] J.-F. Cardoso and A. Solouminac, *Blind beamforming for non-Gaussian signals*, IEE Proceedings-F, Vo. 140, No. 6, pp. 362 – 370, Dec. 1993

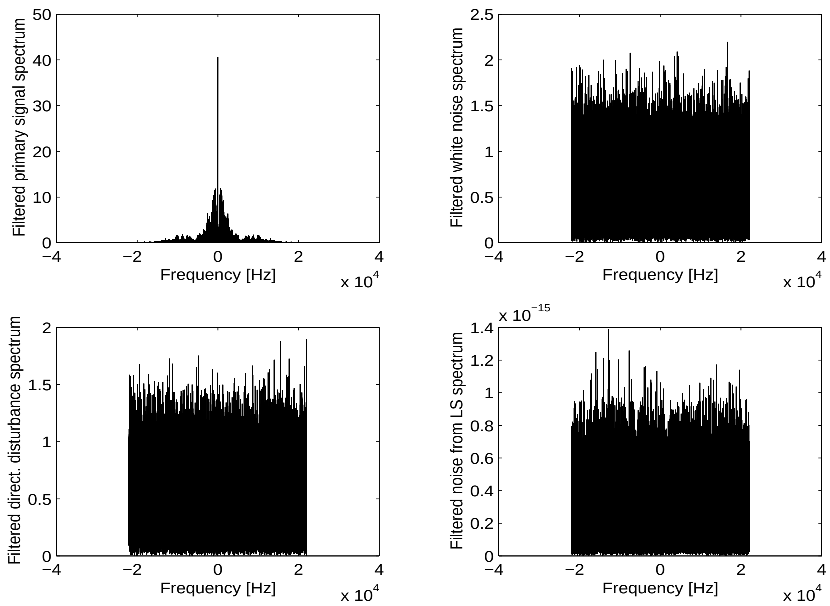

Figure 16: Experiment 3: Spectral contributions to beamformer output in the case $\delta = 0.5$.

[8] S.H. Chang, T.Y. Lee and W.H. Fang, *Hiher-resolution bearing estimation via unitary decomposition artificial neural network*, IEICE Trans. on Fundamentals, Vol. E81A, No. 11, pp. 2455 – 2462, 1998

[9] C. Chatterjee, Z. Kang, and V.P. Roychowdhury, *Algorithms for Accelerated Convergence of Adaptive PCA*, IEEE Trans. on Neural Networks, Vol. 11, No. 2, pp. 338 – 355, March 2000

[10] T.P. Chen, S. Amari, and Q. Lin, *A Unified Algorithm for Principal and Minor Components Extraction*, Neural Networks, Vol. 11, No. 3, pp. 385 – 390, 1998

[11] P.L. Chu, *Superdirective micropone array for a set-top videoconferencing system*, Proc. of International Conference on Acoustics, Speech and Signal Processing, Vol. 1, pp. 235 – 238, 1997

[12] A. Cichocki, W. Kasprzak, and W. Skarbek, *Adaptive Learning Algorithm for Principal Component Analysis with Partial Data*, Proc. Cybernetics and Systems, Vol. 2, pp. 1014 – 1019, 1996

[13] S. Costa and S. Fiori, *Image Compression Using Principal Component Neural Networks*, Image and Vision Computing Journal (special issue on

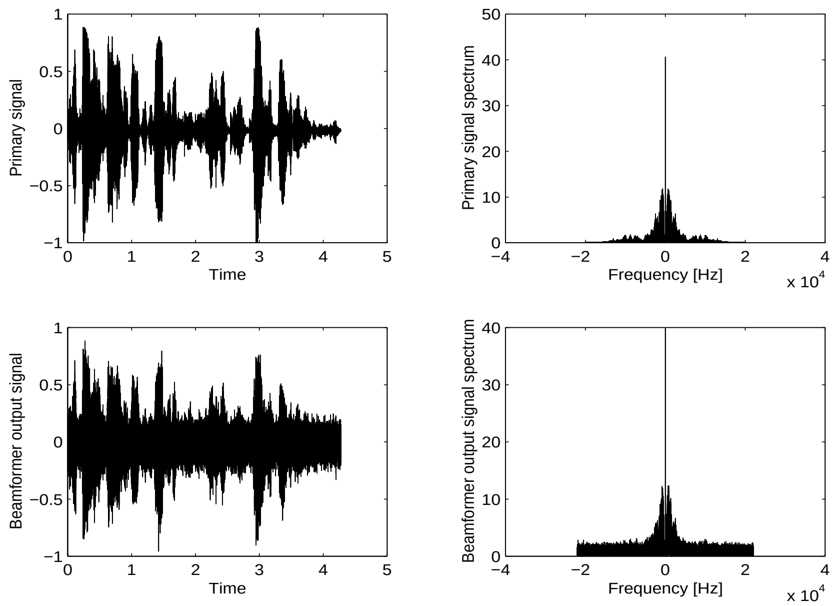

Figure 17: Experiment 3: Time/spectral beamformer output compared to time/spectral primary source in the case $\delta = 0.5$.

"Artificial Neural Network for Image Analysis and Computer Vision"), Vol. 19, No. 9-10, pp. 649 – 668, Aug. 2001

[14] H. Cox, R.M. Zeskind, and M.M. Owen, *Robust adaptive beamforming*, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 10, pp. 1365 – 1376, Oct. 1987

[15] H. Cox, R.M. Zeskind, and T. Kooij, *Practical supergain*, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 3, PP. 393 – 398, Oct. 1986

[16] M.C.F. De Castro, F.C.C. De Castro, J.N. Amaral, and P.R.G. Franco, *A Complex Valued Hebbian Learning Algorithm*, Proc. of International Joint Conference on Neural Networks (IEEE-IJCNN), pp. 1235 – 1238, 1998

[17] K.I. Diamantaras and S.-Y. Kung, *Cross-correlation neural network models*, IEEE Trans. on Signal Processing, Vol. 42, No. 11, Nov. 1994

[18] K.-L. Du, A.K.Y. Lai, K.K.M. Cheng and M.N.S. Swamy, *Neural methods for antenna array signal processing: A Review*, Signal Processing, Vol. 82, pp. 547 – 561, April 2002

[19] K.-L. Du and M.N.S. Swamy, *An iterative blindcyclostationary beamforming algorithm*, Proc. of the IEEE International Conference on Communications, New York City, April-May 2002

[20] A. Hyvärinen and E. Oja, *Independent Component Analysis by General Non-Linear Hebbian-Like Rules*, Signal Processing, Vol. 64, No. 3, pp. 301 – 313, 1998

[21] J. Joutsensalo and J. Karhunen, *A Nonlinear Extension of the Generalized Hebbian Learning*, Neural Processing Letters, Vol. 2, No. 1, pp 5 – 8, 1995

[22] S. Fiori and F. Piazza, *Neural MCA for Robust Beamforming*, Proc. of International Symposium on Circuits and Systems (ISCAS'2000), Vol. III, pp. 614 – 617, May 2000

[23] S. Fiori and F. Piazza, *A General Class of ψ–APEX PCA Neural Algorithms*, IEEE Trans. on Circuits and Systems – Part I, Vol. 47, No. 9, pp. 1394 – 1398, Sept. 2000

[24] S. Fiori, *On Blind Separation of Complex-Valued Sources by Extended Hebbian Learning*, IEEE Signal Processing Letters, Vol. 8, No. 8, pp. 217 – 220, Aug. 2001

[25] S. Fiori, *A Theory for Learning by Weight Flow on Stiefel-Grassman Manifold*, Neural Computation, Vol. 13, No. 7, pp. 1625 – 1647, July 2001

[26] S. Fiori, *Extended Hebbian Learning for Blind Separation of Complex-Valued Sources*, IEEE Trans. on Circuits and Systems – Part II. Accepted for publication

[27] S. Fiori, L. Albini, A. Faba, E. Cardelli and P. Burrascano, *Numerical Modeling for the Localization and the Assessment of Electromagnetic Field Sources*, IEEE Trans. on Magnetics. Accepted for publication

[28] J.L. Flanagan, J.D. Johnston, R. Zhan, and G.W. Elko, *Computer-steered microphone array for sound transduction in large rooms*, Journal of Acoustical Society of America, Vol. 78, No. 5, pp. 1508 – 1518, Nov. 1985

[29] S. Haykin, *Neural Networks*, Ed. MacMillan College Publishing Company, 1994

[30] K. Hornik and C.-M. Kuan, *Convergence Analysis of Local Feature Extraction Algorithm*, Neural Networks, Vol. 5, pp. 229 – 240, 1992

[31] Y.S. Hwu and M.D. Srinath, *A neural network approach to design of smart antennas for wireless communication systems*, Proceeding of the 31th Asilomar Conference on Signals, Systems and Computers, pp. 145 – 148, 1998

[32] J. Karhunen, *Stability of Oja's subspace rule*, Neural Computation, Vol. 6, pp. 739 – 747, 1994

[33] J. Karhunen and J. Joutsensalo, *Representation and Separation of Signals Using Nonlinear PCA Type Learning*, Neural Networks, Vol. 7, No. 1, pp. 113 – 127, 1994

[34] J. Karhunen and J. Joutsensalo, *Generalizations of PCA, Optimization Problems, and Neural Networks*, Neural Networks, Vol. 8, No. 4, pp. 549 – 562, 1995

[35] J.M. Kates, *Superdirective arrays for hearing aids*, Journal of Acoustics Society of America, Vol. 94, No. 4, pp. 1930 – 1933, Oct. 1993

[36] R. Klemm, *Adaptive Airborne MTI: An Auxiliary Channel Approach*, IEE Proceedings F, Vol. 134, pp. 269 – 276, 1987

[37] S.Y. Kung, K.I. Diamantaras and J.S. Taur, *Adaptive principal component extraction (APEX) and applications*, IEEE Trans. on Signal Processing, Vol.42, No.5, pp. 1202 – 1217, May 1994

[38] S. Li and T.J. Sejnowski, *Adaptive separation of mixed broadband sound sources with delays by a beamforming Herault-Jutten network*, IEEE Journal of Oceanic Engineering, Vol. 20, No. 1, January 1995

[39] G. Mathew and V. Reddy, *Orthogonal Eigensubspace Estimation Using Neural Networks*, IEEE Trans. on Signal Processing, Vol. 42, pp. 1803 – 1811, 1994

[40] E. Oja, *A simplified neuron model as a principal component analyzer*, Journal of Mathematics and Biology, Vol. 15, pp. 267 – 273, 1982

[41] E. Oja, *Neural networks, principal components, and subspaces*, International Journal of Neural System, Vol. 1, pp. 61 – 68, 1989

[42] E. Oja, *Principal components, minor components, and linear neural networks*, Neural Networks, Vol. 5, pp. 927 – 935, 1992

[43] E. Oja and L. Wang, *Robust fitting by non-linear neural units*, Neural Networks, Vol. 9, No. 3, pp. 435 – 444, 1996

[44] M.D. Plumbley, *Lyapunov functions for convergence of principal component algorithms*, Neural Networks, Vol. 8, No. 1, pp. 11 – 23, 1995

[45] H. Robbins and S. Monro, *A Stochastic Approximation Method*, Annals of Mathematical Statistics, Vol. 22, pp. 400 – 407, 1951

[46] J. Rubner and P. Tavan, *A self-organizing network for Principal-Component Analysis*, Europhysics Letters, Vol.10, No.7, pp. 693 – 698, 1989

[47] T.D. Sanger, *Optimal unsupervised learning in a single-layer linear feedforward neural network*, Neural Networks, Vol. 2, pp. 459 – 473, 1989

[48] R. Schmidt, *Multiple Emitter Location and Signal Parameter Estimation*, IEEE Trans. on Antennas and Propagation, Vol. 34, pp. 276 – 280, 1986

[49] L. Wang, J. Karhunen and E. Oja, *A bigradient optimization approach for robust PCA, MCA and source separation*, Proc. of International Joint Conference on Neural Network (IEEE-IJCNN), pp. 1684 – 1689, 1995

[50] L. Wiscott, *Learning invariance manifolds*, Proc. of International Conference on Artificial Neural Networks (ICANN), pp. 555 – 560, 1998

[51] L. Xu, *Least Mean Square Error reconstruction principle for self-organizing neural-nets*, Neural Networks, Vol. 6, pp. 627 – 648, 1993

[52] L. Xu, E. Oja, and C.Y. Suen, *Modified Hebbian learning for curve and surface fitting*, Neural Networks, Vol. 5, pp. 441 – 457, 1992

A Proofs of Theorems

A.1 Proof of Theorem 1

Let us expand vector $\mathbf{w}(t)$ by means of the system's eigenbasis [29, 47], that means writing:

$$\mathbf{w}(t) = \theta_1(t)\mathbf{q}_1 + \theta_2(t)\mathbf{q}_2 + \cdots + \theta_p(t)\mathbf{q}_p \ , \tag{22}$$

where the scalar functions $\theta_k(t) \in \mathbb{R}$ are termed "principal modes". Plugging equation (22) into system (6) yields:

$$\sum_{h=1}^{p} \frac{d\theta_h(t)}{dt}\mathbf{q}_h = \sum_{h=1}^{p} \theta_h(t)\mathbf{\Phi}\mathbf{q}_h - \left\{ \sum_{k=1}^{p} [\theta_k(t)\mathbf{q}_k]^T \mathbf{\Phi} \sum_{\ell=1}^{p} [\theta_\ell(t)\mathbf{q}_\ell] \right\} \sum_{h=1}^{p} \theta_h(t)\mathbf{q}_h \ .$$

By recalling the fundamental relationship $\mathbf{\Phi}\mathbf{q}_k = \sigma_k \mathbf{q}_k$ we have:

$$\sum_{h=1}^{p} \frac{d\theta_h(t)}{dt}\mathbf{q}_h = \sum_{h=1}^{p} \theta_h(t)\mathbf{q}_h\sigma_h - \left[\sum_{k=1}^{p}\sum_{\ell=1}^{p} \theta_k(t)\theta_\ell(t)\sigma_\ell \mathbf{q}_k^T\mathbf{q}_\ell\right]\sum_{h=1}^{p}\theta_h(t)\mathbf{q}_h \ .$$

Now from the identity $\mathbf{q}_k^T\mathbf{q}_\ell = \delta_{k\ell}$, it follows that the differential equations for the h^{th} principal modes, with $h \geq 2$, read:

$$\dot{\theta}_h(t) = \theta_h(t)\sigma_h - \sum_{k=1}^{p}\theta_k^2(t)\sigma_k\theta_h(t) \ , \ h = 2, \ldots, p \ , \tag{23}$$

while it is particularly useful to write on a separate equation the differential law pertaining to the first principal mode $\theta_1(t)$, that is:

$$\dot{\theta}_1(t) = \theta_1(t)\sigma_1 - \theta_1^3(t)\sigma_1 - \sum_{k=2}^{p}\theta_k^2(t)\sigma_k\theta_1(t) \ . \tag{24}$$

Our aim is now to solve differential system (23)+(24) that is a coupled non-linear system. In order to have the non-linear differential sub-system (23) decoupled, let us define the new functions:

$$\phi_h(t) \stackrel{\text{def}}{=} \frac{\theta_h(t)}{\theta_1(t)} \ , \ h = 2, \ldots, p \ ,$$

referred to as pseudo-modes; the new function basis $\{\theta_1(t), \phi_2(t), \ldots, \phi_p(t)\}$ for $\mathbf{w}(t)$ allows to equivalently represent the subsystem (23) in a simpler form. To show this property, let us consider that for the h^{th} pseudo-mode there holds:

$$\frac{d}{dt}\phi_h(t) = \frac{\dot{\theta}_h(t)\theta_1(t) - \theta_h(t)\dot{\theta}_1(t)}{\theta_1^2(t)} \ . \tag{25}$$

By using equations (23) and (24) within equation (25), direct calculations show that the dynamics of the pseudo-modes $\phi_h(t)$ look:

$$\frac{d}{dt}\phi_h(t) = (\sigma_h - \sigma_1)\phi_h(t) \ , \ h = 2, \ldots, p \ .$$

thus the principal modes follow the dynamics:

$$\theta_h(t) = \phi_h(0)e^{(\sigma_h - \sigma_1)t}\theta_1(t) \ , \tag{26}$$

The above formula shows that the sub-system (23) has been successfully decoupled, since the dynamics of each principal mode no longer depend upon the other modes, apart from $\theta_1(t)$.

Now, the aim is to solve the differential equation (24) for $\theta_1(t)$. It is worth noting that such differential equation slightly simplifies if we perform the variable change $\theta_1(t) = c(t)e^{\sigma_1 t}$, $c(t) \in \mathbb{R}$. After this we have:

$$\theta_1^2(t) = c^2(t)e^{2\sigma_1 t} \ , \ \theta_h^2(t) = e^{2\sigma_h t}\phi_h^2(0)c^2(t) \ , \ h \geq 2 \ ,$$

then the differential equation (24) becomes:

$$\dot{c}(t) = -c^3(t)e^{2\sigma_1 t}\sigma_1 - \sum_{k=2}^{p} e^{2\sigma_k t}\phi_k^2(0)\sigma_k c^3(t) \ . \tag{27}$$

By defining the following quantity that does not depend on $c(t)$:

$$G(t) \stackrel{\text{def}}{=} -\sum_{k=2}^{p} e^{2\sigma_k t}\sigma_k\phi_k^2(0) - \sigma_1 e^{2\sigma_1 t} \ , \tag{28}$$

the differential equation for $c(t)$ rewrites, compactly:

$$\dot{c}(t) = G(t)c^3(t) \ . \tag{29}$$

Since by definition $G(t) < 0$ and by the hypotheses $c(0) = \mathbf{w}_0^T\mathbf{q}_1 \neq 0$, the above differential equation can be solved by:

$$\int_{c(0)}^{c(t)} \frac{dc}{c^3} = \int_0^t G(\tau)d\tau \ \Rightarrow \ \frac{1}{c(t)^2} = \frac{1}{c(0)^2} - 2\int_0^t G(\tau)d\tau \ . \tag{30}$$

This readily leads to:

$$\frac{1}{\theta_1(t)^2} = \frac{e^{-2\sigma_1 t}}{\theta_1^2(0)} - 2e^{-2\sigma_1 t}\int_0^t G(\tau)d\tau \ .$$

From definition (28) it can be seen that, under condition $\sigma_1 \notin \{\sigma_2, \ldots, \sigma_p\}$, there holds:

$$\lim_{t\to+\infty} 2e^{-2\sigma_1 t}\int_0^t G(\tau)d\tau = -1 \ , \tag{31}$$

from which it is straightforward to conclude that:

$$\lim_{t\to+\infty} \frac{1}{\theta_1^2(t)} = 1 \ . \tag{32}$$

As a consequence, from equation (26) it is readily seen that any $\theta_h(t)$, for $h \neq 1$, asymptotically vanishes to zero provided the hypotheses are met. In this case, the expansion (22) reduces to $\mathbf{w} = \pm\mathbf{q}_1$, which proves the claim.

A.2 Proof of Theorem 2

The proof essentially follows that of Theorem 1, except that now the last principal mode $\theta_p(t)$ plays the central role. Consider the principal modes dynamics:

$$\begin{aligned}
\dot{\theta}_h(t) &= -\theta_h(t)\sigma_h + \sum_{k=1}^{p} \theta_k^2(t)\sigma_k\theta_h(t) \ , \ h = 1, \ldots, p-1 \ , \\
\dot{\theta}_p(t) &= -\theta_p(t)\sigma_p + \sum_{k=1}^{p-1} \theta_k^2(t)\sigma_k\theta_p(t) + \theta_p^3(t)\sigma_p \ .
\end{aligned}$$

Let us define again the pseudo-modes with respect to mode $\theta_p(t)$:

$$\phi_h(t) \stackrel{\text{def}}{=} \frac{\theta_h(t)}{\theta_p(t)} \ , \ h = 1, \ldots, p-1 \ .$$

by means of which it is possible to show that:

$$\theta_h(t) = \phi_h(0) e^{-(\sigma_h - \sigma_p)t} \theta_p(t) \ .$$

Introducing the auxiliary function $c(t)$ such that $\theta_p(t) = c(t)e^{-\sigma_p t}$, we find the resolving differential equation:

$$\dot{c}(t) = c^2(t) e^{-2\sigma_p t} \sigma_p c(t) - \sum_{k=1}^{p-1} e^{-2\sigma_k t} \phi_k^2(0) \sigma_k c^3(t) \ .$$

It is useful to define the function:

$$H(t) \stackrel{\text{def}}{=} \sum_{k=1}^{p-1} e^{-2\sigma_k t} \sigma_k \phi_k^2(0) + \sigma_p e^{-2\sigma_p t} \ ,$$

by means of which it is straightforward to show that $c(t)$ satisfies again equation (29), with function $G(t)$ replaced with $H(t)$, thus the solution has the form (30). Now the time-integral of $H(t)$ is explicitly needed; direct calculations give:

$$-2e^{2\sigma_p t} \int_0^t H(\tau) d\tau = e^{2\sigma_p t} \sum_{k=1}^{p-1} \phi_k^2(0)(e^{-2\sigma_k t} - 1) + 1 - e^{2\sigma_p t} \ ,$$

thus we conclude that:

$$\frac{1}{\theta_p^2(t)} = 1 - \Theta(t) e^{2\sigma_p t} \ , \tag{33}$$

where $\Theta(t) \geq 0$ is an increasing function of the time. It is therefore immediate to find that there exists $\bar{t} \in \mathbb{R}$ such that $\theta_p^{-2}(\bar{t}) = 0$.