Research Article

An Analytical Multimodulus Algorithm for Blind Demodulation in a Time-Varying MIMO Channel Context

Steredenn Daumont^{1,2} and Daniel Le Guennec²

¹IPSS Company, 3 Square du chêne Germain, 35510 Cesson-Sévigné, France

²IETR/Supélec, Campus de Rennes, Avenue de la Boulaie, CS 47601, 35576 Cesson-Sévigné Cedex, France

Correspondence should be addressed to Steredenn Daumont, steredenn.daumont@supelec.fr

Received 30 June 2009; Accepted 5 October 2009

Academic Editor: Hongxiang Li

Copyright © 2010 S. Daumont and D. Le Guennec. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the issue of blind multiple-input multiple-output (MIMO) demodulation of communication signals, with time-varying channels and in an interception context. A new adaptive-blind source separation algorithm, which is based on the implementation of the Multimodulus cost function by analytical methods, is proposed. First a batch processing analysis is performed; then an adaptive implementation of the (Analytical Multi-Modulus Algorithm) AMMA and its simplified version named (Analytical Simplified Constant Modulus Algorithm) ASCMA is detailed. These algorithms, named adaptive-AMMA and adaptive-ASCMA, respectively, are compared with the adaptive (Analytical Constant Modulus Algorithm) ACMA and the MMA (Multi-Modulus Algorithm). The adaptive-AMMA and adaptive-ASCMA achieve a lower residual intersymbol interference and bit error rate than those of the adaptive-ACMA and MMA.

1. Introduction

The context of this study is the Multiple-Input Multiple-Output (MIMO) interception, and more specifically blind demodulation in reception. The channel is supposed to be not stationary, so the methods used to demodulate the received signals must be fast. Such systems exist, among them are the Blind Source Separation (BSS) implemented with analytical methods.

BSS algorithms require few information about emission signals in order to process and recover the transmitted symbols (sources). This type of algorithms estimate an equalization matrix from received signals to obtain the transmitted sources. To find this matrix, different cost function can be minimized, like the Multimodulus (MM) [1, 2], the Constant Modulus (CM) [3, 4], the Simplified Constant Modulus (SCM) [5], and the Multiuser Kurtosis (MUK) [6]. The algorithms in [1–6] use the stochastic gradient to minimize these cost functions, so they are slow to converge and with time-varying environments they are less accurate. So, analytical methods such as the adaptive

Analytical Constant Modulus Algorithm (ACMA) [7–10], which converges quickly, should be used with time-varying channels.

Another particularity of the BSS algorithms is their ability to restore, in the output, the sources with a certain permutation and a certain rotation or phase. Moreover, when a frequency offset is present, the constellations in output of adaptive-ACMA and ACMA spin. In order to avoid that and track the frequency offset jointly to the blind source separation, an MM or SCM cost functions can be used. To our knowledge, the MM and the SCM cost functions associated with analytical methods have never been tried so far (we have presented these works partly in conference [11]). So, in this paper an adaptive Analytical Multimodulus Algorithm and its simplified version named the adaptive Analytical Simplified Constant Modulus Algorithm are proposed.

The performances of the adaptive-AMMA and adaptive-ASCMA are compared with those of the adaptive-ACMA and MMA. With time-selective channels, the adaptive-AMMA and the adaptive-ASCMA allow to obtain a higher signal-to-interference noise ratio (SINR) and a lower bit error rate

(BER), than those obtained with the adaptive-ACMA and the MMA. Moreover, both proposed methods perform a source separation and a carrier phase recovery, unlike the adaptive-ACMA.

First of all, in this paper, the system model used here is described. Then a batch analysis of the blind source separation algorithm AMMA is performed, and an adaptive form of this algorithm and of its simplified version is presented. Finally, simulation results of the proposed algorithms are compared to those obtained with the adaptive-ACMA and MMA in time-varying environments.

The Kronecker and Khatri-Rao products have the following properties:

$$\begin{aligned}\text{vec}(\mathbf{a}\mathbf{b}^H) &= \mathbf{b}^* \otimes \mathbf{a}, \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= \mathbf{AC} \otimes \mathbf{BD}, \\ \text{vec}(\mathbf{ABC}) &= (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}), \\ \text{vec}(\mathbf{A} \text{diag}(\mathbf{b})\mathbf{C}) &= (\mathbf{C}^T \circ \mathbf{A})\mathbf{b}.\end{aligned}$$

2. System Model

A MIMO system with N_t transmitters and N_r receivers is considered. In this system, the information symbols $\mathbf{x}(k)$ are transmitted through the time-varying MIMO $N_r \times N_t$ channel $\mathbf{H}(k)$ at the time k . Taking into consideration a carrier frequency offset δ_f , the $N_r \times 1$ vector of the received signals $\mathbf{y}(k)$ can be expressed as

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k)e^{-j2\pi k\delta_f T_s} + \mathbf{b}(k), \quad (1)$$

where

$$\begin{aligned}\mathbf{y}(k) &= \begin{pmatrix} y_1(k) \\ \vdots \\ y_{N_r}(k) \end{pmatrix}, \quad \mathbf{b}(k) = \begin{pmatrix} b_1(k) \\ \vdots \\ b_{N_r}(k) \end{pmatrix}, \\ \mathbf{H}(k) &= \begin{pmatrix} h_{11}(k) & h_{1N_t}(k) \\ \vdots & \vdots \\ h_{N_r,1}(k) & h_{N_r,N_t}(k) \end{pmatrix}, \quad \mathbf{x}(k) = \begin{pmatrix} x_1(k) \\ \vdots \\ x_{N_t}(k) \end{pmatrix},\end{aligned} \quad (2)$$

and

- (i) $\mathbf{y}$ is an $(N_r \times 1)$ representing the baseband received symbols without any time oversampling,
- (ii) $\mathbf{x}$ is an $(N_t \times 1)$ vector representing the transmitted signals (sources). The sources are independent, identically distributed (i.i.d.), and mutually independent with zero mean and unit variance,
- (iii) $\mathbf{b}$ is an $(N_r \times 1)$ vector representing the additive Gaussian noise,
- (iv) $\mathbf{H}$ is an $(N_r \times N_t)$ unknown, full column rank, instantaneous linear and time-selective MIMO channel,
- (v) $1/T_s$ is the baud rate of the transmitter.

3. The AMMA: A BSS Analytical Method so as to Minimize the MM Cost Function

3.1. Hypotheses. The BSS can be performed under the following hypotheses.

- (i) $N_r \geq N_t$.
- (ii) $\mathbf{H}$ has i.i.d. complex components, is unitary unlike the received signals $\mathbf{y}(k)$, and must be prewhitened before applying the BSS.
- (iii) The noise is additive, white, and Gaussian with a zero mean and a covariance matrix $\mathbf{C}_b = E[\mathbf{b}\mathbf{b}^H] = \sigma_b^2 \mathbf{I}_{N_r}$.
- (iv) The sources are zero mean discrete-time sequences, with a covariance matrix $\mathbf{C}_x = E[\mathbf{x}\mathbf{x}^H] = \mathbf{I}_{N_t}$, and they must be mutually independent at a given time and identically distributed.

3.2. BSS's Principle. The BSS objective is to find a matrix $(N_r \times N_t)\mathbf{W}(k)$ such as

$$\mathbf{z}(k) = \mathbf{W}^H(k)\mathbf{y}(k), \quad (3)$$

where the output vector sequence $\mathbf{z}(k)$ contains accurate estimates of the N_t source signals $\mathbf{x}(k)$. Each column of $\mathbf{W}(k)$ allows to find one source. So, all the columns of $\mathbf{W}(k)$ must be orthogonal with one another, in order to obtain independent sources. The matrix channel $\mathbf{H}$ must be unitary, otherwise the received signals will have to be prewhitened. Generally $\mathbf{H}$ is not unitary, so we will consider a prewhitening operation which also reduces the dimension of $\mathbf{x}(k)$ from N_r to N_t . An underscore is used here to denote pre-whitened variables. Thus, let $\underline{\mathbf{y}}(k) = \mathbf{F}^H \mathbf{y}(k)$ be the prewhitened received signals, where $\mathbf{F}$ is an $(N_r \times N_t)$ prefilter. Like in [8], the prewhitening algorithm in [12] is used. The initial problem to find $\mathbf{W}(k)$ is then replaced by finding a $(N_t \times N_t)$ matrix $\mathbf{T}(k)$:

$$\begin{aligned}\mathbf{z}(k) &= \mathbf{T}^H(k)\underline{\mathbf{y}}(k) \\ &= \mathbf{T}^H(k)\mathbf{F}^H(k)\mathbf{y}(k) \\ &= \mathbf{W}^H(k)\mathbf{y}(k) \\ &= \mathbf{W}^H(k)(\mathbf{H}(k)\mathbf{x}(k) + \mathbf{b}(k)) \\ &= \underline{\mathbf{\Xi}}^H(k)\mathbf{x}(k) + \mathbf{b}'(k),\end{aligned} \quad (4)$$

where $\mathbf{W} = \mathbf{F}\mathbf{T}$ and $\underline{\mathbf{\Xi}} = \mathbf{H}^H(k)\mathbf{F}\mathbf{T}$ are a global separation matrix. Unfortunately, the calculated matrix $\mathbf{T}(k)$ separates the sources, except for a possible permutation of $\mathbf{z}(k)$ and an arbitrary phase on each source signal. The different BSS cost functions, described in Section 3.3, are indeed minimized with any permutation and any rotation on sources. Thus after separation, separator outputs present the following form:

$$\mathbf{z}(k) = \Phi \Gamma \mathbf{x}(k), \quad (5)$$

where Φ is a diagonal matrix and Γ is a permutation matrix which represent the arbitrary phase and the permutation, respectively.

In order to find $\mathbf{T}(k)$, several cost functions can be used. The next section recalls the well-known (Constant Modulus) CM cost function and (Multimodulus) MM cost function.

3.3. Cost Function

3.3.1. CM Cost Function. Many telecommunication signals have a constant modulus (PSK, 4-QAM), normalized to one generally. If the samples of one source are shown on a complex plan, every samples are on an unitary circle, but it is not true if we show a linear combination of several sources. It is this property that cost function CM exploits. Thus, the algorithms using the CM cost function make use of the sources' constant-modulus property:

$$\begin{aligned} J(\mathbf{T}(k)) &= \sum_{n=1}^{N_t} E \left[\left(|z_n(k)|^2 - R \right)^2 \right] \\ &= \sum_{n=1}^{N_t} E \left[\left(\left| \mathbf{t}_n^H(k) \mathbf{y}(k) \right|^2 - R \right)^2 \right], \end{aligned} \quad (6)$$

where $\mathbf{t}_n$ represents the n th row of $\mathbf{T}$ and $R = R_i = R_r = E[|\Re(\mathbf{x}(k))|^4]/E[|\Re(\mathbf{x}(k))|^2]$. This criterion forces BSS outputs to be on a circle with radius R .

3.3.2. MM Cost Function. The cost function is composed of two cost functions; one cost function for real part ($\Re$) and one for imaginary part ($\Im$) of the equalizer output $\mathbf{z}(k)$:

$$\begin{aligned} J(\mathbf{T}(k)) &= \sum_{n=1}^{N_t} \left(E \left[\left(\Re^2(\mathbf{t}_n^H(k) \mathbf{y}(k)) - R_r \right)^2 \right] \right. \\ &\quad \left. + E \left[\left(\Im^2(\mathbf{t}_n^H(k) \mathbf{y}(k)) - R_i \right)^2 \right] \right). \end{aligned} \quad (7)$$

The error estimation for the real and imaginary parts is done jointly, and so the cost function uses implicitly the phase of the equalizer output. Thus, the carrier phase recovery is also accomplished with blind equalization, while the equalizer output signal constellation, obtained when using the CM cost function, suffers from an arbitrary phase rotation. So, the MM cost function is most interesting and used later in this paper.

3.3.3. SCM Cost Function. The SCM cost function [5] consists in projection of the equalizer outputs on one dimension (either real or imaginary part). This cost function presents a lower computational complexity compared to the CM and MM cost functions:

$$J(\mathbf{T}(k)) = \sum_{n=1}^{N_t} E \left[\left(\Re^2(\mathbf{t}_n^H(k) \mathbf{y}(k)) - R_r \right)^2 \right]. \quad (8)$$

3.4. Implementation of Cost Function. Several methods exist to implement cost functions. Among them are the stochastic gradient descent and analytical methods.

The algorithms using stochastic gradient are adaptive algorithms which are slow to converge. Furthermore, in varying environments, subsequent signals tend to be less accurate because of a maladjustment. These algorithms, using a stochastic gradient in order to minimize the cost functions CM and MM, are named, respectively, (Constant Modulus Algorithm) CMA [3, 4] and (MultiModulus Algorithm) MMA [1].

Analytical methods exist in both the adaptive form and batch form. With the adaptive form, the algorithm converges quickly, so it is an interesting option in varying environments.

Our choice is to use, for varying environments, an adaptive analytical method so as to minimize the MM cost function. So, a new algorithm named AMMA and which uses this association, is proposed, once a batch analysis has been performed.

3.5. The Analytical Multimodulus Algorithm. The following cost function should be minimized:

$$J(\tilde{\mathbf{T}}) = \sum_{l=1}^{N_t} E \left[\left(\Re^2(z_l(k)) - R_r \right)^2 + \left(\Im^2(z_l(k)) - R_i \right)^2 \right]. \quad (9)$$

In order to bring out the real and imaginary parts $z_l(k)$, this last term is written as

$$\begin{aligned} z_l(k) &= \mathbf{t}_l^H \mathbf{y}(k), \quad l \in \{1, \dots, N_t\}, \\ &= \tilde{\mathbf{t}}_l^T \tilde{\mathbf{y}}(k) + j \tilde{\mathbf{t}}_l^T \bar{\mathbf{y}}(k), \end{aligned} \quad (10)$$

where $\tilde{\mathbf{t}}_l = \begin{pmatrix} \Re(\mathbf{t}_l) \\ \Im(\mathbf{t}_l) \end{pmatrix}$, $\tilde{\mathbf{y}}(k) = \begin{pmatrix} \Re(\mathbf{y}(k)) \\ \Im(\mathbf{y}(k)) \end{pmatrix}$, $\bar{\mathbf{y}}(k) = \begin{pmatrix} \Im(\mathbf{y}(k)) \\ -\Re(\mathbf{y}(k)) \end{pmatrix}$, $\bar{\mathbf{y}}(k) = \mathbf{G} \tilde{\mathbf{y}}(k)$ with $\mathbf{G} = \begin{pmatrix} \mathbf{0}_{N_t} & \mathbf{I}_{N_t} \\ -\mathbf{I}_{N_t} & \mathbf{0}_{N_t} \end{pmatrix}$.

The matrix $\mathbf{G}$ has the following properties:

$$\begin{aligned} -\mathbf{G} &= \mathbf{G}^T = \mathbf{G}^{-1}, \\ \mathbf{G} \otimes \mathbf{G} &= (\mathbf{G} \otimes \mathbf{G})^T = (\mathbf{G} \otimes \mathbf{G})^{-1}. \end{aligned} \quad (11)$$

Using Kronecker product properties and taking into account [9], $\Re^2(z_n(k))$ and $\Im^2(z_n(k))$ can be written as

$$\begin{aligned} \Re^2(z_l(k)) &= \tilde{\mathbf{t}}_l^T (\tilde{\mathbf{y}}(k) \tilde{\mathbf{y}}^T(k)) \tilde{\mathbf{t}}_l \\ &= (\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k))^T (\tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l), \\ \Im^2(z_l(k)) &= \tilde{\mathbf{t}}_l^T (\bar{\mathbf{y}}(k) \bar{\mathbf{y}}^T(k)) \tilde{\mathbf{t}}_l \\ &= (\bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k))^T (\tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l) \\ &= [(\mathbf{G} \otimes \mathbf{G})(\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k))]^T (\tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l). \end{aligned} \quad (12)$$

3.5.1. The Batch Analysis. Let us denote N a fixed number of received signals, the N rows of $(\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k))^T$ and $(\bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k))^T$ are then, respectively, stacked in the $N \times (2N_t)^2$ matrix $\tilde{\mathbf{P}}$ and $\bar{\mathbf{P}}$.

Using the Khatri-Rao product ($\mathbf{A} \circ \mathbf{B} = (a_1 \otimes b_1 \ a_2 \otimes b_2 \ \dots)$), $\tilde{\mathbf{P}}$ and $\bar{\mathbf{P}}$ can be written as $\tilde{\mathbf{P}} = (\tilde{\mathbf{Y}} \circ \tilde{\mathbf{Y}})^T$ and $\bar{\mathbf{P}} = (\bar{\mathbf{Y}} \circ \bar{\mathbf{Y}})^T$. Let us define the $(2N_t)^2 \times 1$ vectors $\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l$ and $\mathbf{1} = (1, \dots, 1)^T$, the cost function is then given by

$$J(\tilde{\mathbf{T}}) = \frac{1}{N} \sum_{l=1}^{N_t} \left[\|\tilde{\mathbf{P}}\mathbf{d}_l - R \cdot \mathbf{1}\|^2 + \|\bar{\mathbf{P}}\mathbf{d}_l - R \cdot \mathbf{1}\|^2 \right] \quad (13)$$

with $R = R_i = R_r$.

The linear system can be rewritten by using a QR decomposition [7, 8].

We have $\mathbf{Q}$, an $N_s \times N_s$ unitary matrix, and such as $\mathbf{Q}\mathbf{1} = \sqrt{N_s} \begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}$.

After doing the QR factorization of $(\mathbf{1} \ \tilde{\mathbf{P}})$ and $(\mathbf{1} \ \bar{\mathbf{P}})$ and applying $\mathbf{Q}$ to $(\mathbf{1} \ \tilde{\mathbf{P}})$ and $(\mathbf{1} \ \bar{\mathbf{P}})$, we obtain

$$\begin{aligned} \mathbf{Q}(\mathbf{1} \ \tilde{\mathbf{P}}) &= \sqrt{N_s} \begin{pmatrix} 1 & \tilde{\mathbf{y}}^T \\ \mathbf{0} & \tilde{\mathbf{O}} \end{pmatrix}, \\ \mathbf{Q}(\mathbf{1} \ \bar{\mathbf{P}}) &= \sqrt{N_s} \begin{pmatrix} 1 & \bar{\mathbf{y}}^T \\ \mathbf{0} & \bar{\mathbf{O}} \end{pmatrix} \end{aligned} \quad (14)$$

with $\mathbf{1}$ the column vector composed of N_s ones, $\mathbf{0}$ a column vector composed of $N_s - 1$ zeros, $\tilde{\mathbf{y}}^T$ and $\bar{\mathbf{y}}^T$ two $1 \times (2N_t)^2$ rows vectors and $\tilde{\mathbf{O}}$ and $\bar{\mathbf{O}}$ two $N_s \times (2N_t)^2$ matrices, then

$$\begin{aligned} \tilde{\mathbf{P}}\mathbf{d}_l = R\mathbf{1} &\Leftrightarrow \mathbf{Q}(\mathbf{1} \ \tilde{\mathbf{P}}) \begin{pmatrix} -R \\ \mathbf{d}_l \end{pmatrix} = \mathbf{0} \Leftrightarrow \begin{cases} \tilde{\mathbf{y}}^T \mathbf{d}_l = R, \\ \tilde{\mathbf{O}}\mathbf{d}_l = \mathbf{0}, \end{cases} \\ \bar{\mathbf{P}}\mathbf{d}_l = R\mathbf{1} &\Leftrightarrow \mathbf{Q}(\mathbf{1} \ \bar{\mathbf{P}}) \begin{pmatrix} -R \\ \mathbf{d}_l \end{pmatrix} = \mathbf{0} \Leftrightarrow \begin{cases} \bar{\mathbf{y}}^T \mathbf{d}_l = R, \\ \bar{\mathbf{O}}\mathbf{d}_l = \mathbf{0}. \end{cases} \end{aligned} \quad (15)$$

The cost function becomes

$$\begin{aligned} J_{\text{AMMA}}(\mathbf{d}_l) &= \frac{1}{N_s} \sum_{l=1}^{N_t} \left[\left| \tilde{\mathbf{y}}^T \mathbf{d}_l - R \right|^2 + \left| \bar{\mathbf{y}}^T \mathbf{d}_l - R \right|^2 + \|\tilde{\mathbf{O}}\mathbf{d}_l\|^2 + \|\bar{\mathbf{O}}\mathbf{d}_l\|^2 \right], \end{aligned} \quad (16)$$

where $|\tilde{\mathbf{y}}^T \mathbf{d}_l - R|^2$ and $|\bar{\mathbf{y}}^T \mathbf{d}_l - R|^2$ allow to avoid the trivial solution $\mathbf{d}_l = \mathbf{0}$. By squaring (14), the expressions of vectors

$\tilde{\mathbf{y}}$ and $\bar{\mathbf{y}}$, and of matrices $\tilde{\mathbf{C}} = \tilde{\mathbf{O}}^T \tilde{\mathbf{O}}$ and $\bar{\mathbf{C}} = \bar{\mathbf{O}}^T \bar{\mathbf{O}}$ are

$$\begin{aligned} \tilde{\mathbf{y}} &= \frac{1}{N_s} \tilde{\mathbf{P}}^T \mathbf{1} = \frac{1}{N_s} \sum_{k=1}^{N_s} \tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k), \\ \tilde{\mathbf{C}} &= \tilde{\mathbf{O}}^T \tilde{\mathbf{O}} = \frac{1}{N_s} \tilde{\mathbf{P}}^T \tilde{\mathbf{P}} - \frac{1}{N_s} \tilde{\mathbf{P}}^T \mathbf{1} \cdot \frac{1}{N_s} \mathbf{1}^T \tilde{\mathbf{P}} \\ &= \frac{1}{N_s} \sum_{k=1}^{N_s} (\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k)) (\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k))^T \\ &\quad - \frac{1}{N_s^2} \left(\sum_{k=1}^{N_s} \tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k) \right) \left(\sum_{k=1}^{N_s} \tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k) \right)^T, \end{aligned} \quad (17)$$

$$\begin{aligned} \bar{\mathbf{y}} &= \frac{1}{N_s} \bar{\mathbf{P}}^T \mathbf{1} = \frac{1}{N_s} \sum_{k=1}^{N_s} \bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k), \\ \bar{\mathbf{C}} &= \bar{\mathbf{O}}^T \bar{\mathbf{O}} = \frac{1}{N_s} \bar{\mathbf{P}}^T \bar{\mathbf{P}} - \frac{1}{N_s} \bar{\mathbf{P}}^T \mathbf{1} \cdot \frac{1}{N_s} \mathbf{1}^T \bar{\mathbf{P}} \\ &= \frac{1}{N_s} \sum_{k=1}^{N_s} (\bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k)) (\bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k))^T \\ &\quad - \frac{1}{N_s^2} \left(\sum_{k=1}^{N_s} \bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k) \right) \left(\sum_{k=1}^{N_s} \bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k) \right)^T. \end{aligned}$$

By using the definition of the estimated covariance matrices of $\tilde{\mathbf{y}}(k)$ and $\bar{\mathbf{y}}(k)$, named, respectively, $\hat{\mathbf{R}}_{\tilde{\mathbf{y}}}$ and $\hat{\mathbf{R}}_{\bar{\mathbf{y}}}$, and by using the property of the Kronecker product, we obtain the following equalities:

$$\tilde{\mathbf{y}} = \text{vec} \left(\frac{1}{N_s} \sum_{k=1}^{N_s} \tilde{\mathbf{y}}(k) \tilde{\mathbf{y}}^T(k) \right) = \text{vec}(\hat{\mathbf{R}}_{\tilde{\mathbf{y}}}), \quad (18)$$

$$\bar{\mathbf{y}} = \text{vec} \left(\frac{1}{N_s} \sum_{k=1}^{N_s} \bar{\mathbf{y}}(k) \bar{\mathbf{y}}^T(k) \right) = \text{vec}(\hat{\mathbf{R}}_{\bar{\mathbf{y}}}).$$

Thus

$$\begin{aligned} \tilde{\mathbf{y}}^T \mathbf{d}_l &= \text{vec}(\hat{\mathbf{R}}_{\tilde{\mathbf{y}}})^T (\mathbf{t}_l \otimes \mathbf{t}_l) = \mathbf{t}_l^T \hat{\mathbf{R}}_{\tilde{\mathbf{y}}} \mathbf{t}_l, \\ \bar{\mathbf{y}}^T \mathbf{d}_l &= \text{vec}(\hat{\mathbf{R}}_{\bar{\mathbf{y}}})^T (\mathbf{t}_l \otimes \mathbf{t}_l) = \mathbf{t}_l^T \hat{\mathbf{R}}_{\bar{\mathbf{y}}} \mathbf{t}_l. \end{aligned} \quad (19)$$

The criterion J_{AMMA} is minimized when $\tilde{\mathbf{y}}^T \mathbf{d}_l = R$, $\bar{\mathbf{y}}^T \mathbf{d}_l = R$, $\|\tilde{\mathbf{O}}\mathbf{d}_l\|^2 = 0$, and $\|\bar{\mathbf{O}}\mathbf{d}_l\|^2 = 0$. Afterwards, we consider $\tilde{\mathbf{y}}^T \mathbf{d}_l = R$, $\bar{\mathbf{y}}^T \mathbf{d}_l = R$ as constraints. Since received signals are pre-whitened and $\mathbf{R}_{\tilde{\mathbf{y}}} = \mathbf{R}_{\bar{\mathbf{y}}} = \mathbf{I}$ thus

$$\begin{aligned} \tilde{\mathbf{y}}^T \mathbf{d}_l &= \bar{\mathbf{y}}^T \mathbf{d}_l = \mathbf{t}_l^T \mathbf{R}_{\tilde{\mathbf{y}}} \mathbf{t}_l = \mathbf{t}_l^T \mathbf{R}_{\bar{\mathbf{y}}} \mathbf{t}_l \\ &= \mathbf{t}_l^T \mathbf{t}_l = \|\mathbf{t}_l\|^2 = R. \end{aligned} \quad (20)$$

Moreover, $\|\mathbf{d}_l\|^2 = (\mathbf{t}_l \otimes \mathbf{t}_l)^T (\tilde{\mathbf{t}}_l \otimes \mathbf{t}_l) = \|\mathbf{t}_l\|^4 = R^2$. Thus, $\tilde{\mathbf{y}}^T \mathbf{d}_l = \tilde{\mathbf{y}}^T \mathbf{t}_l = R \Leftrightarrow \|\mathbf{d}_l\| = R$, and to a scaling which is not important the optimization problem becomes

$$\begin{aligned} \mathbf{t}_l &= \arg \min_{\substack{\mathbf{d}_l(k)=\mathbf{t}_l(k) \otimes \mathbf{t}_l(k) \\ \|\mathbf{d}_l(k)\|=R}} \sum_{l=1}^{N_t} \left(\|\tilde{\mathbf{O}}\mathbf{d}_l\|^2 + \|\overline{\mathbf{O}}\mathbf{d}_l\|^2 \right) \\ &= \arg \min_{\substack{\mathbf{d}_l(k)=\mathbf{t}_l(k) \otimes \mathbf{t}_l(k) \\ \|\mathbf{d}_l(k)\|=R}} \sum_{l=1}^{N_t} \left(\mathbf{d}_l^T \tilde{\mathbf{O}}^T \tilde{\mathbf{O}} \mathbf{d}_l + \mathbf{d}_l^T \overline{\mathbf{O}}^T \overline{\mathbf{O}} \mathbf{d}_l \right) \\ &= \arg \min_{\substack{\mathbf{d}_l(k)=\mathbf{t}_l(k) \otimes \mathbf{t}_l(k) \\ \|\mathbf{d}_l(k)\|=R}} \sum_{l=1}^{N_t} \mathbf{d}_l^T(k) (\tilde{\mathbf{C}}(k) + \overline{\mathbf{C}}(k)) \mathbf{d}_l(k). \end{aligned} \quad (21)$$

In the appendix, it is shown that vectors $\mathbf{d}_l$, for all $l \in \{1 \dots N_t\}$, which minimize the function $\sum_{l=1}^{N_t} \mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$ under the hypotheses $\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l$ and $\|\mathbf{d}_l\| = R$, are in the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$ (c.f. Theorem 2 and Lemma 2). Moreover, the minimization of $\sum_{l=1}^{N_t} \mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$ is equivalent to minimize $\mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$, for all $l \in \{1 \dots 2\}$ (c.f. Theorem 1 and Lemma 1).

Now, we will see how to create vectors $\mathbf{d}_l$ that present a Kronecker structure, in the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$.

3.5.2. The Batch Method. The vectors $\mathbf{d}_l$ which minimize $\mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$ are eigenvectors associated with the smallest eigenvalues. However, these vectors have not necessarily a Kronecker structure. Thus, in order to obtain the vectors $\tilde{\mathbf{t}}_l$ in the image and to have a Kronecker structure, we search a $N_t \times N_t$ full rank matrix Λ that relates the two bases such so that

$$\tilde{\mathbf{T}} = \arg \min \left\| \mathbf{D} - (\tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}) \Lambda \right\|_F^2 \quad (22)$$

with $\tilde{\mathbf{T}} = (\tilde{\mathbf{t}}_1, \dots, \tilde{\mathbf{t}}_{N_t})$ and $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_{N_t})$ the $2N_t \times N_t$ and $(2N_t)^2 \times N_t$ matrix, respectively. Using the property of the Kronecker product, this minimization problem can be rewritten as

$$\begin{aligned} \left\| \mathbf{D} - (\tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}) \Lambda \right\|_F^2 &= \sum_{n=1}^{N_t} \left\| \mathbf{d}_n - (\tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}) \lambda_n \right\|_F^2 \\ &= \sum_{n=1}^{N_t} \left\| \mathbf{D}_n - \tilde{\mathbf{T}} \Lambda_n \tilde{\mathbf{T}}^T \right\|_F^2 \end{aligned} \quad (23)$$

with

- (i) λ_n the n th vector of the Λ matrix,
- (ii) $\Lambda_n = \text{diag}(\lambda_n)$ is a diagonal matrix,
- (iii) $\mathbf{D}_n = \text{vec}(\mathbf{d}_n)$.

Please note that if the matrix $\tilde{\mathbf{T}}$ was square, the minimization of (23) would be equivalent to a joint diagonalization. In order to obtain the $\tilde{\mathbf{T}}$ matrix, let us suppose in first phase that it is indeed square. Thus, with the aid of joint diagonalization of $\mathbf{D}_n$ matrices, a $2N_t \times 2N_t \tilde{\mathbf{T}}^T$ matrix is obtained. This

diagonalization can be done by using the algorithm described in [13]. The columns of the matrix $\tilde{\mathbf{T}}^T$ are dependent; the $2N_t \times N_t$ matrix $\tilde{\mathbf{T}}$ is extracted by keeping the columns of $\tilde{\mathbf{T}}^T$ linearly independent.

Now, this algorithm is implemented thanks to an adaptive method. So, we must estimate in an adaptive manner, $\tilde{\mathbf{C}}$, $\overline{\mathbf{C}}$, and the eigenvectors of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$ associated with the smallest eigenvalues. The methods used are described in the next section. Then the adaptive tracking of the minor subspace of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$ is performed thanks to the use of the NOOJA algorithm. Finally the adaptive update of the joint diagonalization is realized.

3.5.3. The Adaptive Method. The function $\mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$ will be minimized, at each time k , but iteratively

$$\tilde{\mathbf{t}}_l(k) = \min_{\substack{\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l \\ \|\mathbf{d}_l\| = R}} \mathbf{d}_l^T(k) (\tilde{\mathbf{C}}(k) + \overline{\mathbf{C}}(k)) \mathbf{d}_l(k), \quad \forall l \in \{1, \dots, N_t\}. \quad (24)$$

The matrices $\tilde{\mathbf{C}}$ and $\overline{\mathbf{C}}$ were defined by a batch of N samples. In this adaptive method they are converted into an exponential window (λ -scaling). The details are shown in [8].

The following method used to estimate iteratively the matrix $\tilde{\mathbf{C}}$ and $\overline{\mathbf{C}}$ is a generalization of the Van der Veen method [9]

$$\tilde{\mathbf{C}}(k) = \lambda \tilde{\mathbf{C}}(k-1) + \beta(k) \tilde{\mathbf{c}}(k) \tilde{\mathbf{c}}^T(k) = f(\tilde{\mathbf{C}}(k-1), \tilde{\mathbf{c}}(k)), \quad (25)$$

$$\overline{\mathbf{C}}(k) = \lambda \overline{\mathbf{C}}(k-1) + \beta(k) \overline{\mathbf{c}}(k) \overline{\mathbf{c}}^T(k) = f(\overline{\mathbf{C}}(k-1), \overline{\mathbf{c}}(k)). \quad (26)$$

Then, $\tilde{\mathbf{C}}(k)$ and $\overline{\mathbf{C}}(k)$ are, respectively, the estimations of the autocorrelation matrices for the signals $\tilde{\mathbf{c}}(k)$ and $\overline{\mathbf{c}}(k)$, with $0 < \lambda < 1$, $\beta(k) = (\alpha(k-1)/\alpha(k))\lambda(1-\lambda)$, $\alpha(k) = \lambda\alpha(k-1) + 1 - \lambda$ and

$$\begin{aligned} \tilde{\mathbf{c}}(k) &= \tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k) - \frac{\tilde{\mathbf{p}}(k-1)}{\alpha(k-1)}, \\ \overline{\mathbf{c}}(k) &= \overline{\mathbf{y}}(k) \otimes \overline{\mathbf{y}}(k) - \frac{\overline{\mathbf{p}}(k-1)}{\alpha(k-1)}, \end{aligned} \quad (27)$$

where

$$\begin{aligned} \tilde{\mathbf{p}}(k) &= \lambda \tilde{\mathbf{p}}(k-1) + (1-\lambda) (\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k)), \\ \overline{\mathbf{p}}(k) &= \lambda \overline{\mathbf{p}}(k-1) + (1-\lambda) (\overline{\mathbf{y}}(k) \otimes \overline{\mathbf{y}}(k)). \end{aligned} \quad (28)$$

We saw that the solution of this minimization is in the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$. In order to obtain these vectors in an adaptive manner, a subspace tracking algorithm is used, associated with an adaptive joint diagonalization.

Subspace Tracking. Used alone, the (Normalized Orthogonal Oja) NOOJA algorithm of [14] is utilized to track

minor subspace spanned by the eigenvectors relating to the smallest eigenvalues. But, when associated with an adaptive joint diagonalization, as described in the next section, the NOOJA algorithm tracks the minor subspace spanned by the eigenvectors in the image of $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$ relating to the smallest eigenvalues which are different from 0. This algorithm extracts adaptively the minor subspace $\mathbf{V}$ relating to the input signal $\mathbf{r}(k)$'s autocorrelation matrix $\mathbf{R}$, by maximization the following cost function:

$$\begin{aligned} J(\mathbf{V}) &= E \left\| \mathbf{r}(k) - \mathbf{V} \mathbf{V}^T \mathbf{r}(k) \right\|^2 \\ &= \text{tr}(\mathbf{R}) - 2\text{tr}(\mathbf{V}^T \mathbf{R} \mathbf{V}) + \text{tr}(\mathbf{V}^T \mathbf{R} \mathbf{V} \mathbf{V}^T \mathbf{V}). \end{aligned} \quad (29)$$

In our problem, the minor subspace of $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$ is searched. So, in order to use the NOOJA algorithm, $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$ must be expressed as $\tilde{\mathbf{C}} + \bar{\mathbf{C}} = E[\mathbf{c}\mathbf{c}^T]$. But since the function f (cf. (25) and (26)) is not bilinear, the vector $\mathbf{c}$ is unknown. Instead of searching the minor subspace of $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$, the minor subspace of both $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ will be searched. Since $\bar{\mathbf{C}}$ can not be expressed as

$$\bar{\mathbf{C}} = (\mathbf{G} \otimes \mathbf{G}) \tilde{\mathbf{C}} (\mathbf{G} \otimes \mathbf{G})^{-1}. \quad (30)$$

The matrices $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ are similar. They have therefore the same image. In order to find the minor subspace $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_{N_t})$ spanned by the eigenvectors and relating to the smallest eigenvalues different from 0 for both $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$, we propose to maximize the following cost function:

$$\begin{aligned} J(\mathbf{D}) &= E \left\| \tilde{\mathbf{C}}(k) - \mathbf{D} \mathbf{D}^T \tilde{\mathbf{C}}(k) \right\|^2 + E \left\| \bar{\mathbf{C}}(k) - \mathbf{D} \mathbf{D}^T \bar{\mathbf{C}}(k) \right\|^2 \\ &= \text{tr}(\tilde{\mathbf{C}} + \bar{\mathbf{C}}) - 2\text{tr}(\mathbf{D}^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \mathbf{D}) \\ &\quad + \text{tr}(\mathbf{D}^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \mathbf{D} \mathbf{D}^T \mathbf{D}). \end{aligned} \quad (31)$$

Since $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ have the same image, the first term of the cost function is sufficient to obtain convergence with a constant channel. Using only the matrix $\tilde{\mathbf{C}}$ is equivalent to minimize the cost function Simplified CM [5] in analytical way. This simplification reduces the complexity of the adaptive AMMA and we name it the adaptive Analytical Simplified CMA (ASCMA). Please note that the ASCMA batch analysis is the same with the AMMA batch analysis performed in Section 3.5.1. However, the study is solely about $\tilde{\mathbf{C}}$ rather than $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$.

The maximization of $J(\mathbf{D})$ may be achieved by using the gradient-descent technique:

$$\mathbf{D}(k+1) = \mathbf{D}(k) - \beta \nabla_J(k), \quad (32)$$

where

$$\nabla_J = \left(-2(\tilde{\mathbf{C}} + \bar{\mathbf{C}}) + (\tilde{\mathbf{C}} + \bar{\mathbf{C}}(k)) \mathbf{D} \mathbf{D}^T + \mathbf{D} \mathbf{D}^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \right) \mathbf{D}. \quad (33)$$

The time instant k has been removed to simplify the equations.

By using $\mathbf{D}^T(k) \mathbf{D}(k) = \mathbf{I}$, the cost function can be simplified as

$$J(\mathbf{D}) = \text{tr}(\tilde{\mathbf{C}} + \bar{\mathbf{C}}) - \text{tr}(\mathbf{D}^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \mathbf{D}). \quad (34)$$

By replacing (32) in (34) and by doing the gradient of $J(\mathbf{D}(k+1))$ with respect to $\beta(k)$, the optimal stepsize $\beta_{\text{opt}}(k)$ is obtained

$$\beta_{\text{opt}}(k) = \frac{\text{tr}(\mathbf{D}^T(k) (\tilde{\mathbf{C}}(k) + \bar{\mathbf{C}}(k)) \nabla_J(k))}{\text{tr}(\nabla_J^T(k) (\tilde{\mathbf{C}}(k) + \bar{\mathbf{C}}(k)) \nabla_J(k))}. \quad (35)$$

Keeping in mind that $\mathbf{D}(k)$ is orthogonal at each iteration, that is, $\mathbf{D}^T(k) \mathbf{D}(k) = \mathbf{I}$, and replacing $\tilde{\mathbf{C}}(k)$ and $\bar{\mathbf{C}}(k)$ by the estimate $\tilde{\mathbf{c}}(k) \tilde{\mathbf{c}}^T(k)$ and $\bar{\mathbf{c}}(k) \bar{\mathbf{c}}^T(k)$, respectively, we can write

$$\nabla_J(k) = \tilde{\mathbf{p}}(k) \tilde{\mathbf{z}}^T(k) + \bar{\mathbf{p}}(k) \bar{\mathbf{z}}^T(k), \quad (36)$$

where $\tilde{\mathbf{z}}(k) = \mathbf{D}^T(k) \tilde{\mathbf{c}}(k)$, $\bar{\mathbf{z}}(k) = \mathbf{D}^T(k) \bar{\mathbf{c}}(k)$, $\tilde{\mathbf{p}}(k) = \tilde{\mathbf{z}}(k) - \mathbf{D}(k) \tilde{\mathbf{z}}(k)$, and $\bar{\mathbf{p}}(k) = \bar{\mathbf{z}}(k) - \mathbf{D}(k) \bar{\mathbf{z}}(k)$. The optimal step size becomes

$$\beta_{\text{opt}} = \frac{(\|\tilde{\mathbf{c}}\|^2 - \|\tilde{\mathbf{z}}\|^2) \|\tilde{\mathbf{z}}\|^2 + (\|\bar{\mathbf{c}}\|^2 - \|\bar{\mathbf{z}}\|^2) \|\bar{\mathbf{z}}\|^2}{(\|\tilde{\mathbf{c}}\|^2 - \|\tilde{\mathbf{z}}\|^2)^2 \|\tilde{\mathbf{z}}\|^2 + (\|\bar{\mathbf{c}}\|^2 - \|\bar{\mathbf{z}}\|^2)^2 \|\bar{\mathbf{z}}\|^2}. \quad (37)$$

The instant k has been removed in the $\beta_{\text{opt}}(k)$'s expression in order to simplify the equation.

The updating of $\mathbf{D}(k+1)$ is performed as

$$\mathbf{D}'(k+1) = \mathbf{D}(k) - \beta_{\text{opt}}(k) (\tilde{\mathbf{p}}(k) \tilde{\mathbf{z}}^T(k) + \bar{\mathbf{p}}(k) \bar{\mathbf{z}}^T(k)), \quad (38)$$

where the matrix $\mathbf{D}(k+1)$ is obtained thanks to a Gram-Schmidt orthogonalization of $\mathbf{D}'(k+1)$. The algorithm obtained is described in Algorithm 1.

Once the eigenvectors relating to the smallest eigenvalues of both $\tilde{\mathbf{C}}(k)$ and $\bar{\mathbf{C}}(k)$ are obtained, the constraint $\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l$ must be satisfied. In other words, $\mathbf{d}_l$ must have a Kronecker structure and must be in the image of $\tilde{\mathbf{C}}(k) + \bar{\mathbf{C}}(k)$.

Adaptive Update of the Joint Diagonalization. The method used to satisfy the constraint $\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l$ is done as in [9]. $\tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}$ is computed and regarded as the current estimate of the subspace basis ($\mathbf{D} = \tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}$). Using this basis, the subspace update is performed by using the "Normalized Orthogonal OJA" giving $\mathbf{D}$. The last step of this algorithm is the mapping of the columns $\mathbf{d}_l$ of $\mathbf{D}$ to a Kronecker-product:

$$\begin{aligned} \mathbf{d}_l &= \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l = \text{vec}(\tilde{\mathbf{t}}_l \tilde{\mathbf{t}}_l^T), \\ \text{vec}^{-1}(\mathbf{d}_l) &= \mathbf{D}_l = \tilde{\mathbf{t}}_l \tilde{\mathbf{t}}_l^T. \end{aligned} \quad (39)$$

Then a power iteration [15] is applied. This iteration can be written as

$$\tilde{\mathbf{t}}_l(k+1) = \mathbf{D}_l \tilde{\mathbf{t}}_l(k). \quad (40)$$

At the next iteration, the matrix $\mathbf{D}$ in the image, which has a Kronecker structure, is obtained with $\mathbf{D} = \tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}$. Then, $\mathbf{D}$ is transmitted at the NOOJA algorithm, allowing to vectors $\mathbf{d}_l$ obtained in output NOOJA to be in the nullspace and to have a Kronecker struture.

```

Initialize  $\mathbf{D}(0)$ 
for  $k = 1, 2, \dots$ 
     $\tilde{\mathbf{z}} = \mathbf{D}^T(k)\tilde{\mathbf{c}}, \bar{\mathbf{z}} = \mathbf{D}^T(k)\bar{\mathbf{c}}, \tilde{\mathbf{p}} = \tilde{\mathbf{z}} - \mathbf{D}(k)\tilde{\mathbf{z}}, \bar{\mathbf{p}} = \bar{\mathbf{z}} - \mathbf{V}(k)\bar{\mathbf{z}}$ 
    compute  $\beta_{\text{opt}}$  by using (37)
     $\mathbf{D}'(k+1) = \mathbf{D}(k) - \beta_{\text{opt}}(\tilde{\mathbf{p}}\tilde{\mathbf{z}}^T + \bar{\mathbf{p}}\bar{\mathbf{z}}^T)$ 
     $\mathbf{d}_1(k+1) = \mathbf{d}_1'(k+1)/\|\mathbf{d}_1'(k+1)\|$ 
    for  $q = 2$  to  $N_t$ 
         $\mathbf{s} = [0 \dots 0]^T$ 
        for  $l = 1$  to  $q-1$ 
             $\mathbf{s} = \mathbf{s} + (\mathbf{d}_l^T(k+1)\mathbf{d}_q'(k+1))\mathbf{d}_l(k+1)$ 
        end for
         $\mathbf{d}_q(k+1) = (\mathbf{d}_q'(k+1) - \mathbf{s})/\|(\mathbf{d}_q'(k+1) - \mathbf{s})\|$ 
    end for
end for

```

ALGORITHM 1: Adaptive nullspace of $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ tracking by using NOOJA.

```

Initialize  $\mathbf{F}, \tilde{\mathbf{T}}, \bar{\mathbf{p}} = 0, \tilde{\mathbf{p}} = 0, \alpha = 0$ 
for  $k = 1, 2, \dots$ 
    Update  $\mathbf{F}$ , the prewhitened filter,
    by using  $\mathbf{y}(k)$  and [12] we obtain  $\underline{\mathbf{y}}(k) = \mathbf{F}^H \mathbf{y}(k)$ 
    Update real vectors
     $\tilde{\mathbf{y}}(k) = \begin{pmatrix} \Re(\mathbf{y}(k)) \\ \Im(\mathbf{y}(k)) \end{pmatrix}, \bar{\mathbf{y}}(k) = \begin{pmatrix} -\Im(\mathbf{y}(k)) \\ \Re(\mathbf{y}(k)) \end{pmatrix}, \tilde{\mathbf{w}}_l = \begin{pmatrix} \Re(\mathbf{w}) \\ \Im(\mathbf{w}) \end{pmatrix}$ 
    Update vector  $\tilde{\mathbf{c}}$  and  $\bar{\mathbf{c}}$ 
     $\tilde{\mathbf{c}} = \tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k) - \tilde{\mathbf{p}}/\alpha, \bar{\mathbf{c}} = \bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k) - \bar{\mathbf{p}}/\alpha$ 
     $\tilde{\mathbf{p}} = \lambda\tilde{\mathbf{p}} + (1-\lambda)(\tilde{\mathbf{y}}(k) \otimes \tilde{\mathbf{y}}(k)), \bar{\mathbf{p}} = \lambda\bar{\mathbf{p}} + (1-\lambda)(\bar{\mathbf{y}}(k) \otimes \bar{\mathbf{y}}(k))$ 
     $\alpha = \lambda\alpha + 1 - \lambda$ 
     $\mathbf{D} = \tilde{\mathbf{T}} \circ \tilde{\mathbf{T}}$ 
    Regard  $\mathbf{D}$  as a basis of the nullspace of  $\tilde{\mathbf{C}}$  and  $\bar{\mathbf{C}}$ ,
    and update it using  $\tilde{\mathbf{c}}$  and  $\bar{\mathbf{c}}$  (Algorithm 1)
    for  $l = 1, \dots, N_t$ 
         $\mathbf{D}_l = \text{vec}^{-1}(\mathbf{d}_l), \tilde{\mathbf{t}}_l = \mathbf{D}_l \tilde{\mathbf{t}}_l, \tilde{\mathbf{t}}_l = \tilde{\mathbf{t}}_l / \|\tilde{\mathbf{t}}_l\|$ 
        Obtain complex  $N_t \times N_t$  matrix  $\mathbf{T}(k)$ 
        for  $m = 1, \dots, 2N_t$ 
             $\mathbf{t}_{lm} = \tilde{\mathbf{t}}_{lm} + j\tilde{\mathbf{t}}_{l(m+N_t)}$ 
        end for
    end for
     $\mathbf{T} = \text{recond}(\mathbf{T}), \tilde{\mathbf{T}} = (\Re(\mathbf{T}) \Im(\mathbf{T}))^T$ 
    Estimate the sources
     $\hat{\mathbf{s}}(k) = \mathbf{T}^H(k)\underline{\mathbf{y}}(k)$ 
end for

```

ALGORITHM 2: Adaptive Analytical-MMA.

Estimated Symbols. In order to obtain the transmitted symbols, the complex matrix $\mathbf{T}$ is rebuilt from $\tilde{\mathbf{T}}$

$$\mathbf{t}_{lm} = \tilde{\mathbf{t}}_{lm} + j\tilde{\mathbf{t}}_{l(m+N_t)}, \quad l, m \in 1, \dots, N_t. \quad (41)$$

In order to prevent the extraction of the same source many times, $\mathbf{T}$ is reconditioned after each update. The technique used here is the computation of a singular value

decomposition of $\mathbf{T}$: $\mathbf{T} = \sum \sigma_j \mathbf{u}_j \mathbf{v}_j^H$. Then the singular values of $\mathbf{T}$, which are smaller than 0.5, are replaced by 1

$$\text{recond}(\mathbf{T}) = \sum \sigma'_j \mathbf{u}_j \mathbf{v}_j^H. \quad (42)$$

Then, the matrix $\mathbf{T}$ is applied on the pre-whitened received signals:

$$\hat{\mathbf{s}}(k) = \mathbf{T}^H(k)\underline{\mathbf{y}}(k). \quad (43)$$

The suggested adaptive-AMMA algorithm is described in Algorithm 2.

FIGURE 1: Constellations with a carrier frequency offset. SNR = 30 dB, $N_t = 2$, $N_r = 4$, $\delta_f T_s = 10^{-3}$, $f_d T_s = 0.001$.

4. Simulation Results

The channel used in simulations is a time-selective channel; that is, it is time varying but flat in frequency. The transmitted symbols, resulting from a 4-QAM constellation, are sent through a MIMO Rayleigh fading channel. Two transmitters and four receivers are used ($N_t = 2$ and $N_r = 4$). The performances of proposed adaptive-AMMA are compared with those of the adaptive-ASCMA, the adaptive-ACMA and the MMA, in terms of Bit Error Rate (BER) and Signal to Interference plus Noise Ratio (SINR).

In the simulation results of the Figure 1, we considered the case of a carrier frequency offset ($\delta_f T_s = 10^{-3}$) with SNR = 30 dB. The equalizer output constellation of the adaptive-ACMA (Figure 1(e)) spins because of the carrier frequency offset. Normally, the MMA (Figure 1(f)) allows to track the carrier frequency offset, but in this case the carrier phase is too high to obtain a good tracking. Only the proposed adaptive-AMMA and adaptive-ASCMA can remove the frequency offset (Figures 1(c) and 1(d)), without the use of a carrier tracking loop.

Figure 2 shows the average BER as a function of SNR (with the term “average” being here a time average), and this for two sources and 1000 runs, each one presenting different randomly varying channels and different noise sequences. The value of the Doppler shift $f_d T_s$ is $1 \cdot 10^{-3}$. The simulation here was realized without any carrier frequency offset.

FIGURE 2: BER without carrier frequency offset versus SNR. $N_t = 2$, $N_r = 4$, $\delta_f T_s = 0$, $f_d T_s = 0.0011$.

The lowest BER on Figure 2 is obtained with the proposed adaptive-AMMA algorithm, followed by the adaptive-ASCMA, then by the adaptive-ACMA and the highest BER is obtained with the MMA.

FIGURE 3: Gain of path h_{11} and h_{22} without carrier frequency offset versus symbols number. $N_t = 2$, $\delta_f T_s = 0$, $f_d T_s = 0.0033$ and SNR = 30 dB.

FIGURE 4: SINR without carrier frequency offset versus symbols number. $N_t = 2$, $N_r = 4$, $\delta_f T_s = 0$, $f_d T_s = 0.0033$ and SNR = 15 dB.

So the proposed adaptive-AMMA algorithm allows to obtain a 4 dB gain in SNR terms compared with that obtained by the adaptive-ACMA with a value of BER equal to 10^{-4} .

Figure 3 illustrates the convergence speed of the adaptive-AMMA. The path gain of the channel and its estimate obtained by the adaptive-AMMA are represented on the same figure. We can notice that estimation of adaptive-AMMA follows the variations of the channel.

From the results of the Figure 4, we can see that the performances of the proposed adaptive-AMMA and the

adaptive-ASCMA are very closed in residual SINR terms. Moreover, these two algorithms perform higher residual SINR than the adaptive-ACMA and the MMA.

5. Conclusion

In this paper, we have proposed two new BSS algorithms, named adaptive-AMMA and adaptive-ASCMA, build on the Multimodulus criteria, the Simplified Constant Modulus respectively, and an analytical processing method. The second algorithm is a simplified version of the adaptive-AMMA and it accomplishes performances results closed to the adaptive-AMMA. These algorithms perform better results in BER and SINR terms compared with the adaptive-ACMA and the MMA algorithms. In addition to the blind separation, they can perform a carrier phase recovery simultaneously unlike the adaptive-ACMA.

Appendix

Theorem 1. *The matrices $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ are symmetric and semidefinite positive.*

Proof. Let us consider $\mathbf{A} = \mathbf{I} - (1/N)\mathbf{1}\mathbf{1}^T$ a symmetric matrix, so $\mathbf{A}^T = \mathbf{A}$ and $\mathbf{A}^2 = \mathbf{A}$ thus $\mathbf{A}$ is a projection. Then $\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A}$. Now, we have

$$\begin{aligned}\tilde{\mathbf{C}} &= \frac{1}{N}\tilde{\mathbf{P}}^T\tilde{\mathbf{P}} - \frac{1}{N}\tilde{\mathbf{P}}^T\mathbf{1}\frac{1}{N}\mathbf{1}^T\tilde{\mathbf{P}} \\ &= \frac{1}{N}\tilde{\mathbf{P}}^T\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\tilde{\mathbf{P}} \\ &= \frac{1}{N}\tilde{\mathbf{P}}^T\mathbf{A}\tilde{\mathbf{P}} \\ &= \frac{1}{N}\tilde{\mathbf{P}}^T\mathbf{A}^T\mathbf{A}\tilde{\mathbf{P}} \\ &= \frac{1}{N}[\mathbf{A}\tilde{\mathbf{P}}]^T[\mathbf{A}\tilde{\mathbf{P}}].\end{aligned}\quad (\text{A.1})$$

Similarly

$$\begin{aligned}\bar{\mathbf{C}} &= \frac{1}{N}\bar{\mathbf{P}}^T\bar{\mathbf{P}} - \frac{1}{N}\bar{\mathbf{P}}^T\mathbf{1}\frac{1}{N}\mathbf{1}^T\bar{\mathbf{P}} \\ &= \frac{1}{N}[\mathbf{A}\bar{\mathbf{P}}]^T[\mathbf{A}\bar{\mathbf{P}}].\end{aligned}\quad (\text{A.2})$$

Since $\tilde{\mathbf{C}} = (1/N)[\mathbf{A}\tilde{\mathbf{P}}]^T[\mathbf{A}\tilde{\mathbf{P}}]$ and $\bar{\mathbf{C}} = (1/N)[\mathbf{A}\bar{\mathbf{P}}]^T[\mathbf{A}\bar{\mathbf{P}}]$, then $\tilde{\mathbf{C}}$ and $\bar{\mathbf{C}}$ are symmetric, semidefinite positive and so $\tilde{\mathbf{C}} + \bar{\mathbf{C}}$ too. Furthermore, for all $\mathbf{y} \in \mathbb{R}^{(2N_t)^2}$, $\mathbf{y}^T(\tilde{\mathbf{C}} + \bar{\mathbf{C}})\mathbf{y} \geq 0$. $\square$

Lemma 1. $\mathbf{d}_l^T(\tilde{\mathbf{C}} + \bar{\mathbf{C}})\mathbf{d}_l \geq 0$, for all $l \in \{1 \dots N_t\}$ and so the optimization problem becomes

$$\begin{aligned}\arg \min_{\substack{\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l \\ \|\tilde{\mathbf{d}}_l\| = R}} \sum_{l=1}^{N_t} \mathbf{d}_l^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \mathbf{d}_l &\Leftrightarrow \arg \min_{\substack{\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l \\ \|\tilde{\mathbf{d}}_l\| = R}} \mathbf{d}_l^T (\tilde{\mathbf{C}} + \bar{\mathbf{C}}) \mathbf{d}_l, \\ &\forall l \in \{1 \dots N_t\}.\end{aligned}\quad (\text{A.3})$$

Afterward, let us define the nullspace and the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$ as

$$\begin{aligned} \ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}}) &= \left\{ \mathbf{y} \in \frac{\mathbb{R}^{(2N_t)^2}}{(\tilde{\mathbf{C}} + \overline{\mathbf{C}})\mathbf{y} = \mathbf{0}_{\mathbb{R}^{(2N_t)^2}}} \right\}, \\ \text{image}(\tilde{\mathbf{C}} + \overline{\mathbf{C}}) &= \left\{ \mathbf{z} \in \frac{\mathbb{R}^{*(2N_t)^2}}{\mathbf{z} = (\tilde{\mathbf{C}} + \overline{\mathbf{C}})\mathbf{y}}, \mathbf{y} \in \mathbb{R}^{(2N_t)^2} \right\}, \end{aligned} \quad (\text{A.4})$$

$\ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$ and $\text{image}(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$ are subspaces of $\mathbb{R}^{(2N_t)^2}$.

Property 1. $\ker(\mathbf{C}) \oplus \text{image}(\mathbf{C}) = \mathbb{R}^{(2N_t)^2}$ where $\oplus$ denotes the direct sum, that is, $\ker(\mathbf{C}) \cap \text{image}(\mathbf{C}) = \{\mathbf{0}\}$ and $\ker(\mathbf{C})$ and $\text{image}(\mathbf{C})$ are subspaces of $\mathbb{R}^{(2N_t)^2}$.

Let us assume now, for the sake of simplicity, that $N_t = 2$ and $Nr \geq 2$, $\dim(\tilde{\mathbf{C}} + \overline{\mathbf{C}}) = (2N_t)^2 \times (2N_t)^2 = 16 \times 16$.

Afterward, the following affirmations will be proven. The vectors $\mathbf{d}_l$ which minimize $\mathbf{d}_l^T (\tilde{\mathbf{C}} + \overline{\mathbf{C}}) \mathbf{d}_l$ are in the nullspace of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$, but the sole vectors $\mathbf{0}_{\mathbb{R}^{16}}$ verify the constraint $\mathbf{d}_l = \tilde{\mathbf{t}}_l \otimes \tilde{\mathbf{t}}_l$, yet $\mathbf{0}_{\mathbb{R}^{16}}$ is a trivial solution to be avoided. However, the vectors $\mathbf{d}_l$ are in the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$.

Definition 1. By construction, the sole dependence obtained on the columns of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$ is $C_2 = C_5, C_3 = C_9, C_4 = C_{13}, C_7 = C_{10}, C_8 = C_{14}, C_{12} = C_{15}$, where C_n represents the n th column of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$. So, $\dim[\ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}})] = 6$. A basis $B_{\ker}$ of the nullspace is defined as

$$\begin{aligned} B_{\ker} &= \{e'_1, e'_2, e'_3, e'_4, e'_5, e'_6\} \\ &= \{e_2 - e_5, e_3 - e_9, e_4 - e_{13}, e_7 - e_{10}, e_8 - e_{14}, e_{12} - e_{15}\}, \end{aligned} \quad (\text{A.5})$$

where $e_i, i \in \{1, \dots, 16\}$ are vectors in the standard basis of $\mathbb{R}^{16}$.

The e'_i are linearly independent and verify $(\tilde{\mathbf{C}} + \overline{\mathbf{C}})e'_i = \mathbf{0}$. Thanks to the following rank-nullity theorem:

$$\text{rank}(\tilde{\mathbf{C}} + \overline{\mathbf{C}}) + \dim(\ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}})) = \dim(\mathbb{R}^{16}), \quad (\text{A.6})$$

and since the dimension of $\ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$ is equal to 6, the dimension of $\text{Image}(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$ is equal to 10. And B_{im} is a basis of image:

$$\begin{aligned} B_{im} &= \{e''_1, e''_2, e''_3, e''_4, e''_5, e''_6, e''_7, e''_8, e''_9, e''_{10}\} \\ &= \{e_1, e_6, e_{11}, e_{16}, e_2 + e_5, e_3 + e_9, e_4 + e_{13}, e_7 + e_{10}, \\ &\quad e_8 + e_{14}, e_{12} + e_{15}\}. \end{aligned} \quad (\text{A.7})$$

The e''_i are linearly independent; verify $\mathbf{C}e''_i \neq \mathbf{0}$ and we can verify the Property 1. The e'_i and the e''_i are linearly independent, then $\{e'_1, \dots, e'_6, e''_1, \dots, e''_{10}\}$ is a basis for $\mathbb{R}^{16}$.

Now, the following theorem can be proved.

Theorem 2. In the nullspace of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$, the vector $\mathbf{0}_{\mathbb{R}^{16}}$ is the sole vector to have a Kronecker structure.

Proof. Let $\mathbf{x} \in \mathbb{R}^{16}$, $\mathbf{x} = (x_1, \dots, x_{16})^T$. Supposing $\mathbf{x} \in \ker(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$,

$$\mathbf{x} = \sum_{i=1}^6 \alpha_i e'_i, \quad \alpha_i \in \mathbb{R}. \quad (\text{A.8})$$

We have

$$\begin{aligned} \text{(i)} \quad &x_1 = x_6 = x_{11} = x_{16} = 0, \\ \text{(ii)} \quad &x_2 = -x_5, x_3 = -x_9, x_4 = -x_{13}, x_7 = -x_{10}, x_8 = \\ &-x_{14}, x_{12} = -x_{15}. \end{aligned}$$

So, $\mathbf{X} = \text{vec}^{-1}(\mathbf{x})$ is a (4×4) skew-symmetric matrix.

$\mathbf{X}$ and $\text{vec}^{-1}(\mathbf{t} \otimes \mathbf{t})$ with $\mathbf{t} = (t_1 \ t_2 \ t_3 \ t_4)^T \in \mathbb{R}^4$ are equal to

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} 0 & -x_2 & -x_3 & -x_4 \\ x_2 & 0 & -x_7 & -x_8 \\ x_3 & x_7 & 0 & -x_{12} \\ x_4 & x_8 & x_{12} & 0 \end{pmatrix}, \\ \text{vec}^{-1}(\mathbf{t} \otimes \mathbf{t}) &= \begin{pmatrix} t_1^2 & t_1 t_2 & t_1 t_3 & t_1 t_4 \\ t_1 t_2 & t_2^2 & t_2 t_3 & t_2 t_4 \\ t_1 t_3 & t_2 t_3 & t_3^2 & t_3 t_4 \\ t_1 t_4 & t_2 t_4 & t_3 t_4 & t_4^2 \end{pmatrix}. \end{aligned} \quad (\text{A.9})$$

$\text{vec}^{-1}(\mathbf{t} \otimes \mathbf{t})$ is a (4×4) symmetric matrix and its diagonal values are positive, while $\mathbf{X}$ is a skew-symmetric matrix. So, vectors $\mathbf{x}$ do not have a Kronecker structure, except $\mathbf{0}_{\mathbb{R}^{16}}$. $\square$

So, vectors $\mathbf{d}_l$ are not in the nullspace of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$, but vectors $\mathbf{d}_l$ are in the image of $\tilde{\mathbf{C}} + \overline{\mathbf{C}}$. Using property 1, we deduce the following lemma.

Lemma 2. Let us consider E , the set of vectors in $\mathbb{R}^{16}$, which have a Kronecker structure: $E = \{\mathbf{p} \in \mathbb{R}^{16} / \mathbf{p} = \mathbf{t} \otimes \mathbf{t}, \mathbf{t} \in \mathbb{R}^4\}$, then $E \subset \text{Image}(\tilde{\mathbf{C}} + \overline{\mathbf{C}})$.

Notations

The following notations are adopted:

- *: Complex conjugation
- T : Matrix or vector transpose
- H : Matrix or vector complex conjugate transpose
- $\Re$: Real part of complex
- $\Im$: Imaginary part of complex
- $\mathbf{1}$: Vector of all 1s
- $\mathbf{0}_N$: An $N \times N$ matrix with zeros components
- $\mathbf{I}_N$: The $N \times N$ identity matrix
- $E[\cdot]$: Expectation operator

$\text{diag}(\mathbf{a})$: A diagonal matrix constructed from a vector $\mathbf{a}$

$\text{vec}(\mathbf{A})$: Stacking of the columns of $\mathbf{A}$ into a vector

$\otimes$: Kronecker product

$\circ$: Khatri-Rao product:

$$\mathbf{A} \circ \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots)$$

$\|\cdot\|_F^2$: The Frobenius norm

- [15] G. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 1996.

References

- [1] P. Sansrimahachai, D. Ward, and A. Constantinides, "Blind source separation for BLAST," *Digital Signal Processing*, vol. 1, pp. 139–142, 2002.
- [2] K. N. Oh and Y. O. Chin, "Modified constant modulus algorithm: blind equalization and carrier phase recovery algorithm," in *Proceedings of the IEEE International Conference on Communications*, vol. 1, pp. 498–502, 1995.
- [3] L. Castedo, C. J. Escudero, and A. Dapena, "A blind signal separation method for multiuser communications," *IEEE Transactions on Signal Processing*, vol. 45, no. 5, pp. 1343–1348, 1997.
- [4] R. Gooch and J. Lundell, "The CM array: an adaptive beamformer for constant modulus signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '86)*, vol. 11, pp. 2523–2526, Tokyo, Japan, April 1986.
- [5] A. Ikhlef and D. Le Guennec, "A simplified constant modulus algorithm for blind recovery of MIMO QAM and PSK signals: a criterion with convergence analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Article ID 90401, 13 pages, 2007.
- [6] C. B. Papadias, "Globally convergent blind source separation based on a multiuser kurtosis maximization criterion," *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3508–3519, 2000.
- [7] A.-J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996.
- [8] J. Li and P. Stoica, *Robust Adaptive Beamforming*, Wiley InterScience, New York, NY, USA, 2005.
- [9] A.-J. van der Veen, "An adaptive version of the algebraic constant modulus algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 4, pp. 873–876, Philadelphia, Pa, USA, 2005.
- [10] A.-J. van der Veen, "Asymptotic properties of the algebraic constant modulus algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1796–1807, 2001.
- [11] S. Daumont and D. Le Guennec, "Adaptive analytical MMA with time-varying MIMO channels and diversity in interception context," in *Proceedings of the 16th International Conference on Digital Signal Processing (DSP '09)*, 2009.
- [12] S. Douglas, "Combined subspace tracking, prewhitening, and contrast optimization for noisy blind signal separation," in *Proceedings of the 2nd International Workshop on Independent Component Analysis*, pp. 579–584, 2000.
- [13] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, 1996.
- [14] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Processing Letters*, vol. 8, no. 7, pp. 203–206, 2001.