EUSIPCO 2013 1569746561

SLIDING WINDOW ADAPTIVE CONSTANT MODULUS ALGORITHM BASED ON COMPLEX HYPERBOLIC GIVENS ROTATIONS

A. Boudjellal[1] *K. Abed-Meraim*[1] *A. Belouchrani*[2] *Ph. Ravier*[1]

[1] Polytech'Orleans, Prisme Laboratory, France; [2] Ecole Nationale Polytechnique, Algiers, Algeria.

ABSTRACT

This paper proposes a new adaptive Constant Modulus Algorithm (CMA) for the blind separation of communication signals. Although many existing CMA-like algorithms have been proposed in the literature, their efficiency in terms of convergence rate and separation quality is still relatively low. We introduce herein a new adaptive technique based on the use of complex Hyperbolic Givens rotations which shows very good performance as illustrated by the simulation results and comparative study provided at the end of the paper.

***Index Terms*—** Constant Modulus Algorithm (CMA), Adaptive CMA, Hyperbolic Givens Rotations.

1. INTRODUCTION

The Constant Modulus Algorithm is one of the most efficient techniques for blind equalization or blind separation of communications signals. A plethora of CMA-like methods and algorithms have been proposed so far in the literature, e.g. [3, 4, 10, 11] for the blind equalization and [5, 7, 8, 9] for the blind source separation, to improve and extend the original version introduced more than three decades ago [1, 2]. In particular, many adaptive algorithms have been proposed for the minimization of the non-linear Constant Modulus (CM) cost function. Among the most efficient implementations one can cite the LS-CMA algorithm in [6, 8] and the adaptive Analytical CMA (ACMA) in [9].

Here, we consider another type of adaptive implementations using sliding window and Hyperbolic Givens (HG) rotations. The proposed algorithm has the advantage of fast convergence and good separation quality for a moderate computational cost comparable to that of the methods in [6, 8, 9]. The first part of the paper is dedicated to the algorithm's development while the second part is for the simulation results and comparative study with the LS-CMA and adaptive ACMA methods.

2. PROBLEM FORMULATION

In the context of Multi-user MIMO communications system, the instantaneous mixture $\mathbf{x}_k$ of d transmitted sources $\mathbf{s}_k$ received through an M-antenna array is modelled as follows:

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k. \quad (1)$$

were $\mathbf{A}$ is the $M \times d$ mixing matrix and the M-dimensional vector $\mathbf{n}_k$ is an additive noise of covariance $\sigma_n^2 I$. By stacking K samples of the received data in one matrix $\mathbf{X}$, the model (1) becomes:

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N} \quad (2)$$

where $\mathbf{X} = [\mathbf{x}_1 : \mathbf{x}_K]$, $\mathbf{S} = [\mathbf{s}_1 : \mathbf{s}_K]$ and $\mathbf{N} = [\mathbf{n}_1 : \mathbf{n}_K]$.

Blind Source Separation aims to recover the unknown sources from observed mixtures, relying only on some assumptions on the statistical properties of the original sources [1]. This is equivalent to find a $d \times M$ separation matrix $\mathbf{W}$ which output is the estimated source vector (up to scaling and permutation ambiguities [17]):

$$\mathbf{Z} = \mathbf{W}\mathbf{X} = \widehat{\mathbf{S}}. \quad (3)$$

In the sequel, we'll consider the square case where $M = d$ (one can use signal subspace projection in the non square case where $M > d$ as in [9]).

3. HYPERBOLIC-GIVENS CONSTANT MODULUS ALGORITHM HG-CMA

Originally, the CMA in [1, 2] was designed in such a way it exploits the fact that the original sources are generated from a finite alphabet having a constant modulus[2] R and try to restore this property by minimizing the deviation of the restored signals modulus from this constant leading to the following Constant Modulus Criterion (CMC):

$$\mathcal{J}(\mathbf{W}) = \sum_{i=1}^{M}\sum_{j=1}^{K}\left(|z_{ij}|^2 - R\right)^2 \quad (4)$$

[1] Standard hypotheses consist of assuming that: (i) The mixing matrix $\mathbf{A}$ is a tall full column rank matrix ($M \geq d$), (ii) The original sources are mutually independent, (iii) The additive noise is white, Gaussian, and independent from the source signals.

[2] Later on, it is shown that the CMA can be applied for any sub-Gaussian sources [16]. Also, because of the scaling ambiguity, one can chose $R = 1$, without loss of generality.

where z_{ij} is the $(i,j)^{th}$ entry of the output matrix $\mathbf{Z}$. The minimization of (4) leads to a large number of algorithms belonging to the CMA class. In particular, the authors in [12] proposed a two-step iterative Jacobi-like algorithm for the minimization of the CMC (or the $4-th$ order cumulants in [13]) after data pre-whitening. Adaptive implementation of such two-steps method results, in general, in algorithms of relatively slow convergence rates, e.g. [13]. In [14], a one step joint diagonalization algorithm is proposed by decomposing the separation matrix $\mathbf{W}$ as a product of hyperbolic Givens rotations. We propose here to exploit this matrix decomposition in our context for an adaptive optimization of the CMC.

In other words, at time instant t, the current estimate of the separation matrix $\mathbf{W}^{(t-1)}$ is updated as:

$$\mathbf{W}^{(t)} = \mathbf{D}^{(pq)}(\lambda_p, \lambda_q)\, \mathbf{G}^{(pq)}(\theta, \alpha)\, \mathbf{H}^{(pq)}(\phi, \beta)\, \mathbf{W}^{(t-1)} \tag{5}$$

where:

- $H^{(pq)}(\phi, \beta)$ is the complex non-unitary hyperbolic matrix with diagonal elements equal to one except for the two elements $h_{pp} = h_{qq} = \cosh(\phi)$ and its off-diagonal elements are null except the elements $h_{pq} = h_{qp}^* = e^{i\beta}\sinh(\phi)$.
- $\mathbf{G}^{(pq)}(\theta, \alpha)$ is the complex unitary Givens matrix with diagonal elements equal to one except for the two elements $g_{pp} = g_{qq} = \cos(\theta)$ and its off-diagonal elements are null except for the elements $g_{pq} = -g_{qp}^* = e^{i\alpha}\sin(\theta)$.
- $\mathbf{D}^{(pq)}(\lambda_p, \lambda_q)$ is a normalization diagonal matrix with diagonal elements equal to one except for the two elements $d_{pp} = \lambda_p$, and $d_{qq} = \lambda_q$.

Next, for given rotation indices (p, q), we show how the parameters $\theta, \alpha, \phi, \beta, \lambda_p$ and λ_q are optimized to minimize the CMC.

3.1. Complex Non-Unitary Hyperbolic Rotations

The first stage of the algorithm consists of applying the non-unitary hyperbolic transformation to the received bloc of data[3]

$$\mathbf{Z} = \mathbf{H}^{(pq)}(\phi, \beta)\, \mathbf{X} \tag{6}$$

The couple of hyperbolic angles (ϕ, β) is calculated so that it minimizes the CMC in (4). In fact, this transformation affects only the rows of index p and q of the data bloc $\mathbf{X}$.

It can be shown that, in this case, the CMC can be decomposed into two parts; one of them is a function of (ϕ, β) and the second is independent of these two parameters (referred to as 'constant'):

[3]For the sake of simplicity, we use notations $\mathbf{X}$ for the input data, $\mathbf{u}$ for the parameter vector and $\mathbf{Z}$ for the output data, whatever the considered input and matrix transformation are.

$$\mathcal{J}_H(\phi, \beta) = 2\left(\mathbf{u}^T\mathbf{Q}_s\mathbf{u} - 2R\mathbf{u}^T\mathbf{r}_s\right) + \text{constant} \tag{7}$$

where:

- $\mathbf{u} = [\cosh(2\phi),\ \sinh(2\phi)\cos(\beta),\ \sinh(2\phi)\sin(\beta)]^T$,
- $\mathbf{Q}_s = \sum_{j=1}^{K} \mathbf{r}^{(j)}\mathbf{r}^{(j)\,T}$ and $\mathbf{r}_s = \sum_{j=1}^{K} \mathbf{r}^{(j)}$,
- $\mathbf{r}^{(j)} = \left[\frac{1}{2}\left(|x_{pj}|^2 + |x_{qj}|^2\right),\ \mathcal{R}(x_{pj}\, x_{qj}^*),\ \mathcal{I}(x_{pj}\, x_{qj}^*)\right]^T$,
- x_{ij} is the $(i,j)^{th}$ entry of matrix $\mathbf{X}$.

$\mathcal{R}(.)$ and $\mathcal{I}(.)$ being the real and imaginary parts of a complex number.

Considering the particular structure of vector $\mathbf{u}$, the minimization of (7) is equivalent to the following constrained optimization problem:

$$\min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{Q}_s\mathbf{u} - 2R\,\mathbf{u}^T\mathbf{r}_s\right\},\ \ s.t.\ \ \mathbf{u}^T\mathbf{J}_3\mathbf{u} = 1 \tag{8}$$

where the diagonal matrix $\mathbf{J}_3 = diag([1, -1,\ -1])$ is used to express the particular structure of vector $\mathbf{u}$.

The exact solution of (8) can be obtained using Lagrange multipliers as shown in [15]. However, a simpler approximate solution[4] can be obtained by considering the first order Taylor expansion of the hyperbolic functions around zero leading to (calculation details are omitted due to space limitation):

$$\begin{aligned} \beta &= \text{atan}\left(\frac{\sum_{j=1}^{K} r_3^{(j)}(r_1^{(j)}-1)}{\sum_{j=1}^{K} r_2^{(j)}(r_1^{(j)}-1)}\right) \\ \phi &= \frac{1}{2}\text{atanh}\left(\frac{\sum_{j=1}^{K}\left(\cos(\beta)r_2^{(j)}+\sin(\beta)r_3^{(j)}\right)\left(1-r_1^{(j)}\right)}{\sum_{j=1}^{K}\left(\cos(\beta)r_2^{(j)}+\sin(\beta)r_3^{(j)}\right)^2+\left(r_2^{(j)}-r_1^{(j)}\right)}\right) \end{aligned} \tag{9}$$

where $r_i^{(j)}$ refers to the i-th entry of vector $\mathbf{r}^{(j)}$.

3.2. Complex Unitary Givens Rotations

The second step of the algorithm consists of applying the unitary Givens transformation to the considered data bloc:

$$\mathbf{Z} = \mathbf{G}^{(pq)}(\theta, \alpha)\, \mathbf{X} \tag{10}$$

Like for the hyperbolic transformation, the Givens rotation affects only the rows of index p and q of X. Again, the CMC in (4) can be decomposed into two parts; one of them is a function of (θ, α) and the second is independent of these two parameters:

$$\mathcal{J}_G(\theta, \alpha) = 2\left(\mathbf{u}^T\mathbf{Q}_g\mathbf{u}\right) + \text{constant} \tag{11}$$

where:

[4]This solution has been chosen since it allows us to reduce the computational cost but also because we observed in our simulation experiments that it leads to almost the same algorithm's performance as if we use the exact solution of (8).

- $\mathbf{u} = [cos(2\theta),\ sin(2\theta)cos(\alpha),\ sin(2\theta)sin(\alpha)]^T$,
- $\mathbf{Q}_g = \sum_{j=1}^{K} \mathbf{r}^{(j)}\mathbf{r}^{(j)\ T}$,
- $\mathbf{r}^{(j)} = \left[\frac{1}{2}\left(|x_{pj}|^2 - |x_{qj}|^2\right),\ \mathcal{R}(x_{pj}x_{qj}^*),\ \mathcal{I}(x_{pj}x_{qj}^*)\right]^T$,

Hence, the optimal couple of angles (θ, α) that minimizes the quadratic form in (11) under the constraint $\mathbf{u}^T\mathbf{u} = 1$ is given by:

$$\left\{\begin{array}{lcl} \cos(\theta) & = & \sqrt{(u_1+1)/2} \\ e^{i\alpha}\sin(\theta) & = & (u_2 + iu_3)\,/\sqrt{2(u_1+1)} \end{array}\right. \quad (12)$$

where $\mathbf{u} = [u_1,\ u_2,\ u_3]^T$ is the least eigenvector of $\mathbf{Q}_g$ associated to its smallest eigenvalue[5].

3.3. Normalization Rotations

It has been shown in the two previous subsection that both Givens and hyperbolic transformations affect only the rows of index p and q of the data bloc $\mathbf{X}$. The last algorithm's transform is a normalization step according to

$$\mathbf{Z} = \mathbf{D}^{(pq)}(\lambda_p, \lambda_q)\,\mathbf{X} \quad (13)$$

The optimal parameters (λ_p, λ_q) are calculated so that they minimize the CMC which is expressed, in this case, as (constant terms are omitted):

$$\begin{array}{rcl} \mathcal{J}_D(\lambda_p, \lambda_q) & = & \sum_{j=1}^{K}(\lambda_p^4\,|x_{pj}|^4 - 2R\lambda_p^2\,|x_{pj}|^2) \\ & + & \sum_{j=1}^{K}(\lambda_q^4\,|x_{qj}|^4 - 2R\lambda_q^2\,|x_{qj}|^2) \end{array} \quad (14)$$

Optimal normalization parameters can be obtained at the zeros of the derivatives of (14) with respect to these two parameters:

$$\left\{\begin{array}{lcl} \lambda_p & = & \sqrt{R\,\sum_{j=1}^{K}|x_{pj}|^2\,/\sum_{j=1}^{K}|x_{pj}|^4} \\ \lambda_q & = & \sqrt{R\,\sum_{j=1}^{K}|x_{qj}|^2\,/\sum_{j=1}^{K}|x_{qj}|^4} \end{array}\right. \quad (15)$$

3.4. Adaptive Implementation of HG-CMA

To make an adaptive version of this algorithm, let us consider a sliding bloc $\mathbf{X}^{(t-1)} = [\mathbf{x}_{t-K}, ..., \mathbf{x}_{t-2}, \mathbf{x}_{t-1}]$ which is updated at the acquisition of a new sample $\mathbf{x}_t$ (at time instant t) to be $\mathbf{X}^{(t)} = [\mathbf{x}_{t-K+1}, ..., \mathbf{x}_{t-1}, \mathbf{x}_t]$.

The main idea of the adaptive HG-CMA is to apply one sweep of complex rotations on the sliding window at each time instant and update the separation matrix $\mathbf{W}$ by this sweep of rotations. The resulting algorithm is summarized in Table 1 (for simplicity, we use the same notation for the data and its transformed version).

[5] We have here a 3×3 eigenvalue problem that can be solved explicitly.

```
Initialization: W^(K) = I_d.
For t = K+1, K+2, ... do
    x_t = W^(t-1) x_t
    X^(t) = [x_{t-K}, ..., x_{t-1}, x_t]
    W^(t) = W^(t-1)
    For all 1 ≤ p < q ≤ d do
        Compute H^(pq) using (9)
        Compute G^(pq) using (12)
        Update W^(t) = G^(pq) H^(pq) W^(t)
        Update X^(t) = G^(pq) H^(pq) X^(t)
    end For
    For 1 ≤ p ≤ d,   compute λ_p using (15)  end For
    Compute D = diag([λ_1, ··· , λ_d])
    Normalize W^(t) = D W^(t) and X^(t) = D X^(t)
end For
```

Table 1. Adaptive HG-CMA Algorithm.

Note that, the normalization step is done outside the sweep loop which reduces slightly the numerical cost.

The numerical cost of the HG-CMA is of order $O(d^2K)$ (assuming $K > d$) but can be reduced to $O(dK)$ flops per iteration if we use only one or two rotations per time instant. In the simulation experiments, we compare the perfomance of the algorithm in the 3 following cases:

- When we use one complete sweep (i.e. $d(d-1)/2$ rotations)
- When we use one single rotation which indices are chosen according to an automatic selection (i.e. automatic incrementation) throughout the iterations in such a way all search directions (i.e all indices values) are visited periodically.
- When we use 2 rotations per iteration (time instant): one pair of indices is selected according to the maximum deviation criterion:

$$(p,q) = arg\max\sum_{k=1}^{K}(|x_{pk}|^2 - R)^2 + (|x_{qk}|^2 - R)^2$$

  while the other rotation indices are selected using the previous automatic selection procedure.

Comparatively, the adaptive ACMA [9] costs approximately $O(d^3)$ flops per iteration and the LS-CMA[6] costs $O(d^2K + d^3)$. Interestingly, as shown in section 4, the sliding window length K can be chosen of the same order as the number of sources d without affecting much the algorithm's performance. In that case, the numerical cost of HG-CMA becomes similar to that of the adaptive ACMA.

[6] We consider here an adaptive version of the LS-CMA using the same sliding window as for our algorithm.

4. SIMULATION RESULTS

To assess the performance of the HG-CMA, we consider here a 5×5 MIMO system (i.e. $d = 5$). The inputs are i.i.d 4-PSK modulated sequences and the system matrix $\mathbf{A}$ is generated randomly at each Monte Carlo run but with controlled conditioning (its entries are generated as i.i.d. Gaussian variables). Unless stated otherwise, the processing window size is set to $K = 2d$.

The separation quality is measured by the Signal to Interference and Noise Ratio (SINR) averaged over 200 Monte Carlo runs.

In figure 1, we compare the convergence rates and separation quality of HG-CMA (with different number of rotations per time instant), LS-CMA and adaptive ACMA. One can observe that HG-CMA outperforms the two other algorithms in this simulation context. Also, one can see that even with 2 rotations per time instant, our algorithm leads to high separation quality with fast convergence rate (typically, few tens of iterations are sufficient to reach the steady state level).

In figure 2, the plots represent the steady state SINR (obtained after 1000 iterations) versus the SNR. One can see that the HG-CMA has no saturation effect (as for the LS-CMA and adaptive ACMA) and its SINR increases almost linearly with the SNR in dB.

In figure 3, the SNR is set to $20dB$ and the plots represent again the steady state SINR versus the number of sources d. Severe performance degradation is observed (when the number of sources increases) for the LS-CMA and adaptive ACMA while the HG-CMA performance seems to be unaffected when the source number increases.

In figure 4, the plots illustrate the algorithms performance versus the chosen processing window size[7] K. Surprisingly, HG-CMA algorithm reaches its optimal performance with relatively short window sizes (K can be chosen of the same order as d).

In the last experiment, we consider 8-QAM sources (with non constant modulus property). In that case, all algorithms performance are degraded but HG-CMA still outperforms the two other algorithms.

To improve the performance in the non constant modulus signal case, one needs to increase the processing window size as illustrated by this simulation result but more importantly, one needs to use more elaborated cost functions which combines the CMC with alphabet matching criteria e.g. [10, 11].

5. CONCLUSION

A new adaptive constant modulus algorithm has been introduced using sliding window and hyperbolic Givens rotation. The proposed algorithm is of moderate complexity but has the advantage of fast convergence rate and high separation quality. The simulation results illustrate its effectiveness as compared to the adaptive implementations of ACMA and LS-CMA. They show, in particular, that the sliding window size can be chosen as small as twice the number of sources without significant performance loss. Also, they illustrate the trade off between the convergence rate and the algorithm's numerical cost as a function of the number of used rotations per iteration. The proposed iterative technique can be adapted for the optimization of more elaborated cost functions which combine the CMC with alphabet matching criteria, e.g. [10, 11].

[7] This concerns only LS-CMA and HG-CMA as the adaptive ACMA in [9] uses an exponential window with parameter $\beta = 0.995$.

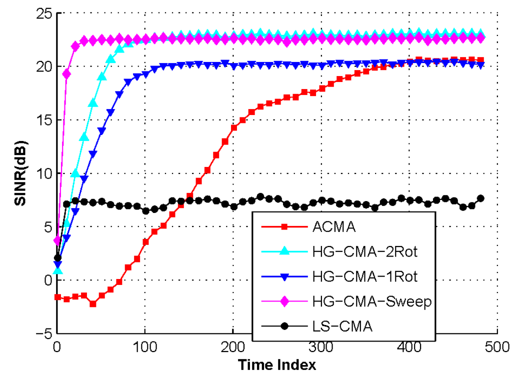

Fig. 1. SINR vs.Time Index: $SNR = 20dB$, $d = 5$, $K = 10$.

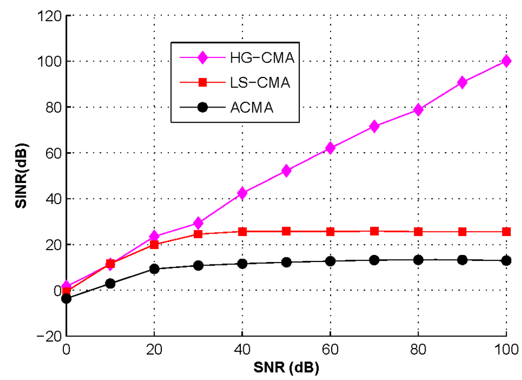

Fig. 2. SINR vs. SNR: $d = 5$, $K = 10$.

6. REFERENCES

[1] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems", *IEEE Trans. Commun.*, vol. COM-28, no. 11, pp. 1867- 1875, Nov. 1980.

[2] J. R. Treichler and B. G. Agee, "A New Approach to the Multipath Correction of Constant Modulus Signals", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 2, pp. 459-471, Apr. 1983.

[3] V. Y. Yang, and D. L. Jones, "A Vector Constant Modulus Al-

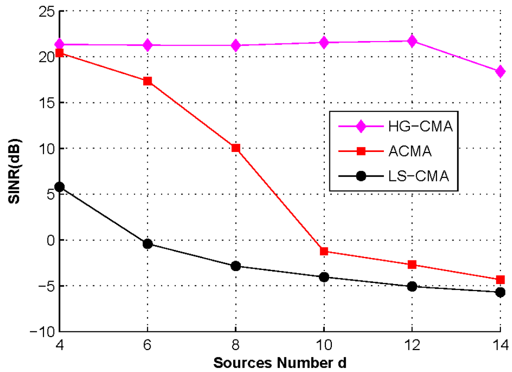

Fig. 3. SINR vs. Source Number: $SNR = 20dB$, $K = 2d$.

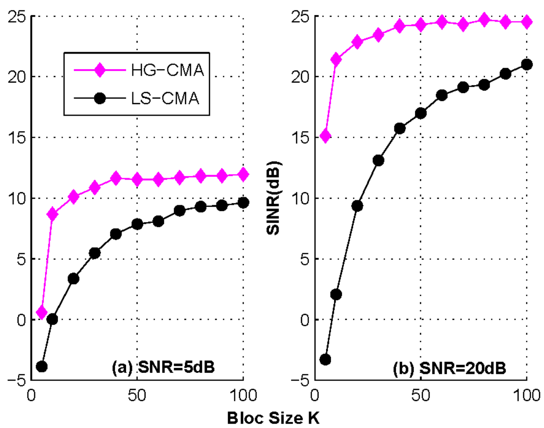

Fig. 4. SINR vs. Bloc Size K: $d = 5$.

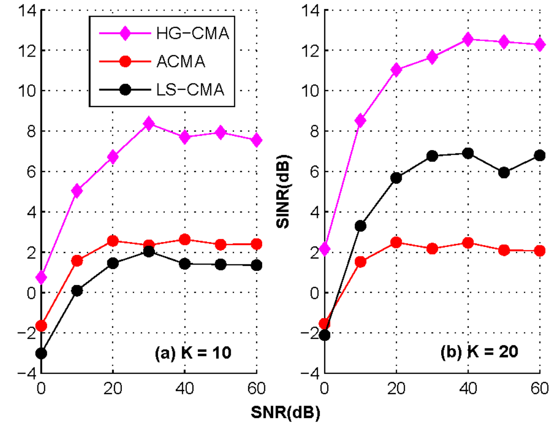

Fig. 5. SINR vs. SNR: $d = 5$.

gorithm for Shaped Constellation Equalization", *IEEE Signal Process. Lett.*, vol. 5, no. 4, pp. 89-91 Apr. 1998.

[4] S. Abrar, A. K. Nandi, "An Adaptive Constant Modulus Blind Equalization Algorithm and its Stochastic Stability Analysis", *IEEE Signal Process. Lett.*, vol. 17, no. 1, pp. 55-58, Jan. 2010.

[5] A. Belouchrani and K. Abed-Meraim, "Constant Modulus Blind Source Separation: A New Approach", *in Proc. ISSPA'96*, Gold Coast, Australia, August 1996.

[6] B. G. Agee, "The Least-Squares CMA: A New Technique for Rapid Correction of Constant Modulus Signals", *In Proc. IEEE ICASSP*, Tokyo, pp. 953-956, 1986.

[7] C. B. Papadias, "Globally Convergent Blind Source Separation Based on a Multiuser Kurtosis Maximization Criterion (2000)", *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3508-3519, Dec. 2000.

[8] C. B. Papadias and A. Kuzminskiy, "Blind Source Separation with Randomized Gram-Shmidt Orthogonalization for Short-Burst Systems", *in Proc. ICASSP'04*, Montreal, Quebec, Canada, May 17-21, 2004.

[9] A. J. Van Der Veen and A. Leshem, "Constant Modulus Beamforming", Chapter 6 in Robust Adaptive Beamforming, (J. Li & P. Stoica, eds.), Wiley Interscience, pp. 299-351, 2005.

[10] Lin He, M. G. Amin, C. Reed, Jr., R. C. Malkemes, "A Hybrid Adaptive Blind Equalization Algorithm for QAM Signals in Wireless Communications", *IEEE Trans. on Signal Proc.*, Vol. 52, No. 7, pp. 2058-2069, July 2004.

[11] A. Labed Abdenour, T. Chonavel, A. Aissa-El-Bey, A. Belouchrani, "Min-Norm Based Alphabet-Matching Algorithm for Adaptive Blind Equalization of High-Order QAM Signals", *European Trans.on Telecom.*, Feb. 2013.

[12] A. Ikhlef, K. Abed-Meraim, D. Guennec, "On the Constant Modulus Criterion: A New Algorithm", *in Proc. ICC'10*, pp. 1-5, May 2010.

[13] M. Thameri, A. Kammoun, K. Abed-Meraim, A. Belouchrani, "Fast Principal Component Analysis and Data Whitening Agorithms", *in Proc. WOSSPA'11*, Tipaza, Algeria, May 9-11, 2011.

[14] A. Souloumiac, "Nonorthogonal Joint Diagonalization by Combining Givens and Hyperbolic Rotations", *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2222-2231, Jun. 2009.

[15] R. Iferroudjene, K. Abed-Meraim, A. Belouchrani, "A New Jacobi-Like Method for Joint Diagonalization of Arbitrary non-Defective Matrices", *Applied Mathematics and Computation* 211(2): 363-373 (2009).

[16] P. A. Regalia, "On the Equivalence Between the Godar and Shalvi-Weistein Schemes of Blind Equalization", *Signal Processing*, vol. 73, pp. 185-190, Feb. 1999.

[17] P. Comon, C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press Inc, 2009.