ADAPTIVE BLIND SPARSE SOURCE SEPARATION BASED ON SHEAR AND GIVENS ROTATIONS

Nacerredine Lassami[⋆] *Abdeldjalil Aïssa-El-Bey*[⋆] *Karim Abed-Meraim*[†]

[⋆] IMT Atlantique, UMR CNRS 6285 Lab-STICC, UBL, 29238 Brest, France
[†]Université d'Orléans, PRISME, 45067 Orléans, France

ABSTRACT

This paper addresses the problem of adaptive blind sparse source separation in the time domain of an over-determined instantaneous noisy mixture. A two-step approach is proposed: first, the data are projected on the signal subspace estimated using the principal subspace tracker FAPI. In the second step, an ℓ_1 criterion is used to represent the sparsity property of the signal sources. For the optimization of this cost function, an adaptive method based on Givens and Shear rotations is used. This algorithm, referred to SGDS-FAPI, guarantees low computational complexity which is essential in the adaptive context. Numerical simulations have been performed, and showed that the proposed algorithm outperforms existing solutions in both convergence speed and estimation quality.

Index Terms— Blind source separation, sparse signals, adaptive algorithm, Shear and Givens rotations

1. INTRODUCTION

Blind source separation (BSS) is a signal processing technology which has been intensively used recently in several areas [1], such as biomedical engineering [2], audio (music and speech) processing [3] and communication applications [4]. The main objective of source separation is to recover unknown transmitted source signals from the observations received at a set of sensors. In BSS neither the sources nor the mixing matrix are known, i.e. it exploits only the information carried by the received signals and a prior information about the statistics or the nature of transmitted source signals (e.g. decorrelation, independence, morphological diversity, etc). Recently, sparsity has emerged as a novel and effective source of diversity for BSS [5–7]. Although the sparse source separation can be particularly useful for separating under-determined mixtures (more sources than sensors), it is also potentially interesting for the noisy over-determined mixture (more sensors than sources) in which case sparsity is exploited to improve the source separation quality [1].

In this paper, we address the problem of adaptive blind sparse source separation in the noisy over-determined case. We introduce the SGDS-FAPI[1] algorithm which is based on a two-step approach as the DS-OPAST algorithm which we have proposed earlier in [8]. However, in this work, the used subspace tracker is more accurate and the sparsity criterion is optimized by means of Givens and Shear (hyperbolic givens) rotations. The Givens-based (Jacobi-like) techniques are attractive due to their numerical stability, their facility to be parallelized and their low computational cost. Such techniques have been already used in the context of BSS [9, 10] but with other a prior information than sparsity.

The rest of the paper is organized as follows: In Section 2, the data model and a brief overview of the considered BSS problem are presented. Derivation of the proposed algorithm is presented in Section 3. Simulation results are presented in Section 4 and Section 5 concludes the paper.

2. DATA MODEL AND PROBLEM FORMULATION

In this paper, we consider the over-determined linear instantaneous mixture model with sparse sources. Let $\mathbf{x}(t) = [x_1(t), \ldots, x_d(t)]^T$ be a random data vector observed at the tth snapshot over an array of d sensors where $(.)^T$ stands for transpose operator. The measured array output is a weighted superposition of the signals, corrupted by additive noise which satisfies the model:

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

where $\mathbf{s}(t)$ is the unknown $p \times 1$ sparse signal vector with a non singular second order covariance $\mathrm{E}[\mathbf{s}(t)\mathbf{s}^H(t)]$ and $(.)^H$ stands for Hermitian transpose operator. In the over-determined case we have more sensors than source signals ($p < d$). $\mathbf{A}$ is the $d \times p$ unknown full rank mixing matrix and $\mathbf{n}(t)$ is a zero-mean additive random white noise. In this paper, we assumed that all the vectors and matrices have real values for the simplicity of the equations. In case of complex values problem, we can either extend the derived equation or simply transfer it to its real equivalent form using the following equation:

$$\begin{pmatrix} \mathcal{R}(\mathbf{x}(t)) \\ \mathcal{I}(\mathbf{x}(t)) \end{pmatrix} = \begin{pmatrix} \mathcal{R}(\mathbf{A}) & -\mathcal{I}(\mathbf{A}) \\ \mathcal{I}(\mathbf{A}) & \mathcal{R}(\mathbf{A}) \end{pmatrix} \begin{pmatrix} \mathcal{R}(\mathbf{s}(t)) \\ \mathcal{I}(\mathbf{s}(t)) \end{pmatrix} \tag{2}$$

[1]SGDS stands for Shear Givens based Data Sparse

where $\mathcal{R}(.)$ and $\mathcal{I}(.)$ represent the real and the imaginary operators for complex values. The latter can be combined with the structure preserving technique shown in [11]. In order to consider the separation in an adaptive context, we assume also a time-varying mixing matrix $\mathbf{A}$ (we omit the time index (t) to reduce the amount of notation).

Solving the blind source separation problem means to find a $p \times d$ separation matrix $\mathbf{B}$ (or equivalently, identifying $\mathbf{A}$ and applying its pseudo-inverse $\mathbf{A}^{\#}$) such that $\hat{\mathbf{s}}(t) = \mathbf{B}\mathbf{x}(t)$ is an estimation of the source signals. Note that complete blind identification of separating matrix is possible only up to permutation and scaling ambiguity i.e. $\mathbf{B}$ is a solution if:

$$\mathbf{B}\mathbf{x}(t) = \mathbf{P}\mathbf{\Lambda}\mathbf{s}(t) \tag{3}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Lambda}$ is a non-singular diagonal matrix.

3. PROPOSED ALGORITHM

We propose a two-step approach to solve the problem following the same procedure as presented in [8]. The main idea is to track the principal subspace of the data in first step using FAPI [12, 13] algorithm. This step is equivalent to a dimension reduction step which maximizes the variance of the projected data into the signal subspace. In the second step, we seek the rotation matrix that transforms the principal subspace matrix into the desired mixing matrix by using the sparsity property of the sources.

3.1. First step

In the first step, our aim is to track the principal subspace matrix $\mathbf{W}(t)$ which should span the same subspace as the one spanned by the p dominant eigenvectors i.e. which correspond to the p greater eigenvalues of the covariance matrix $\mathbf{C}_x = \mathrm{E}[\mathbf{x}(t)\mathbf{x}^H(t)]$. Either exponential or truncated (sliding) data window can be used to adaptively update the estimation of $\mathbf{C}_x(t)$ according to the following qualitative criterion:

• One opts for the exponential window in the case of slowly changing signal parameters since it tends to smooth the variations of the desired parameters. In this case:

$$\mathbf{C}_x(t) = \sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{x}(i)^T = \beta\,\mathbf{C}_x(t-1) + \mathbf{x}(t)\mathbf{x}(t)^T \tag{4}$$

where $0 < \beta \leq 1$ is the forgetting factor.

• The truncated (sliding) window is preferred for faster tracking of signal parameter changes, but it leads to a higher computational complexity and needs more memory than the exponential window.

$$\begin{aligned}\mathbf{C}_x(t) &= \sum_{i=t-L+1}^{t} \beta^{t-i}\,\mathbf{x}(i)\mathbf{x}(i)^T \\ &= \beta\,\mathbf{C}_x(t-1) + \mathbf{x}(t)\mathbf{x}(t)^T - \beta^{l}\mathbf{x}(t-L)\mathbf{x}^T(t-L)\end{aligned} \tag{5}$$

where L is the width of the window.

The FAPI algorithm [12, 13] resolves the problem of tracking the signal subspace of dimension $p < d$ under the orthogonality constraint of the weighting matrix $\mathbf{W}(t)$ for both types of windows. The choice of FAPI was encouraged by its linear complexity and capability to guarantee the orthonormality of the subspace weighting matrix $\mathbf{W}(t)$ at each time step. In fact, FAPI has one of the best trade-off between quality of estimation and complexity of calculation (for more details see [12, 13]).

3.2. Second step

The FAPI output matrix $\mathbf{W}(t)$ should also span the same subspace as our mixing matrix $\mathbf{A}$ and we can write that

$$\mathbf{A} = \mathbf{W}(t)\mathbf{Q}(t) \tag{6}$$

where $\mathbf{Q}(t)$ is a non singular square matrix. Note also that finding the matrix $\mathbf{A}^{\#}$ (with permutation and scaling ambiguity) is somehow equivalent to finding $\mathbf{B}$ the separation matrix. Therefore, in this second step the non singular matrix $\mathbf{Q}(t)$ is introduced in order to optimize the criterion which describes the sparsity of the separated sources. The most natural way to present the sparsity of a signal is the ℓ_0 pseudo norm which counts the number of the non-zero coefficients. Unfortunately, the ℓ_0 makes the objective function non convex, non continuous and hard to optimize (requiring time-consuming solutions which are not suitable for our adaptive case). Therefore, we use the relaxation to ℓ_1 norm which is one of the best convex approximation of ℓ_0 pseudo norm [14]. Hence, the objective function considered to restore the sparsity of the source signals (estimated as $\mathbf{A}^{\#}\mathbf{x}(t)$) is given by:

$$\begin{aligned}\mathrm{J}(\mathbf{Q}(t)) &= \|(\mathbf{W}(t)\mathbf{Q}(t))^{\#}\mathbf{X}(t)\|_1 \\ &= \|\mathbf{Q}(t)^{-1}\mathbf{W}(t)^T\mathbf{X}(t)\|_1\end{aligned} \tag{7}$$

where $\mathbf{X}(t) = [\beta^{L-1}\mathbf{x}(t-L+1), \beta^{L-2}\mathbf{x}(t-L+2), \ldots, \mathbf{x}(t)]$ is the windowed data matrix (exponentially when $L = t$ and truncated when $L < t$). In order to optimize the above criterion w.r.t $\mathbf{Q}(t)$, we propose to write the matrix $\mathbf{Q}(t)^{-1}$ as a product of elementary Givens and Shear matrices:

$$\mathbf{Q}(t)^{-1} = \prod_{1 \leq i < j \leq p} \mathcal{S}_{ij}\mathcal{G}_{ij} \tag{8}$$

Indeed, any non singular matrix can be decomposed into product of Shear $\mathcal{S}_{ij}$ and Givens $\mathcal{G}_{ij}$ elementary matrices (up to a constant factor) for $1 \leq i < j \leq p$ which are defined as an identity matrix except for their $(i,i)^{th}$, $(i,j)^{th}$, $(j,i)^{th}$ and $(j,j)^{th}$ entries given by:

$$\begin{bmatrix}\mathcal{G}_{ij}(i,i) & \mathcal{G}_{ij}(i,j)\\ \mathcal{G}_{ij}(j,i) & \mathcal{G}_{ij}(j,j)\end{bmatrix} = \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{bmatrix} \tag{9}$$

$$\begin{bmatrix}\mathcal{S}_{ij}(i,i) & \mathcal{S}_{ij}(i,j)\\ \mathcal{S}_{ij}(j,i) & \mathcal{S}_{ij}(j,j)\end{bmatrix} = \begin{bmatrix}\cosh(\phi) & \sinh(\phi)\\ \sinh(\phi) & \cosh(\phi)\end{bmatrix} \tag{10}$$

where θ and ϕ are the Givens angle and the Shear parameter. Next, we present the case of using just one Shear-Givens rotation every time iteration for simplicity and then we generalize to the case where we consider more than one rotation. Hence, we can write that

$$\mathbf{Q}^{-1}(t) = \mathbf{H}(t) = \mathcal{S}_{ij}\mathcal{G}_{ij}\mathbf{H}(t-1) \tag{11}$$

for the selected indices (i, j) at the t^{th} time iteration.

In order to minimize the objective function $J(\mathbf{Q}(t))$, one needs to specify how the rotation indices are chosen at each iteration as well as how the parameters (θ, ϕ) are optimized. We start by proposing an automatic selection strategy for the rotation indices (i.e. automatic incrementation) throughout the iterations in such a way all search indices values are visited periodically. Hence, if (l, m) are the rotation indices at time instant $t-1$, then at the current time instant, we will have:

$$(i,j) = \begin{cases} (l, m+1) & \text{if} \quad m < p \\ (l+1, l+2) & \text{if} \quad m = p \quad \text{and} \quad l < p-1 \\ (1, 2) & \text{if} \quad m = p \quad \text{and} \quad l = p-1 \end{cases}$$

which allows us to scan all the $p(p-1)/2$ possible values. Hence, after fixing the indices (i, j), finding $\mathbf{H}(t)$ resumes in estimating the parameters (θ, ψ) which minimize:

$$J(\theta,\psi) = \|\mathcal{S}_{ij}\mathcal{G}_{ij}\mathbf{H}(t-1)\mathbf{W}^T(t)\mathbf{X}(t)\|_1 \tag{12}$$

$$= \left\| \begin{bmatrix} \cosh(\phi) & \sinh(\phi) \\ \sinh(\phi) & \cosh(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} \right\|_1 \tag{13}$$

where $\mathbf{y}_i$ and $\mathbf{y}_j$ are the i^{th} and j^{th} rows of the product $\mathbf{H}(t-1)\mathbf{W}^T(t)\mathbf{X}(t)$ available from the previous iteration. We can expand the equation to:

$$\begin{aligned} J(\theta,\psi) = & \|(\cos(\theta)\cosh(\psi) - \sin(\theta)\sinh(\psi))\mathbf{y}_i + \\ & (\cos(\theta)\sinh(\psi) + \sin(\theta)\cosh(\psi))\mathbf{y}_j\|_1 + \\ & \|(\cos(\theta)\sinh(\psi) - \sin(\theta)\cosh(\psi))\mathbf{y}_i + \\ & (\cos(\theta)\cosh(\psi) + \sin(\theta)\sinh(\psi))\mathbf{y}_j\|_1 \end{aligned} \tag{14}$$

$J(\theta, \psi)$ is a scalar function of two variables with no simple analytic solution for its minimum point, so we have used a MATLAB numerical search function called "fminsearch" which finds the minimum of unconstrained multivariate function using derivative-free method.

Remark: It is possible to optimize separately the Givens parameter θ then the Shear parameter ψ. This leads to a slight loss in terms of estimation quality but helps reducing the computational cost since one replaces the 2D search by two 1D parameter optimization.

3.3. Combining both steps

Now, we will resume the scheme to follow in order to reach a linear computational complexity for both steps. The main idea is to run the FAPI algorithm independently and to update the separation matrix

$$\mathbf{B}(t) = \mathbf{H}(t)\mathbf{W}^T(t) \tag{15}$$

at every iteration. Note that Eq. (14) includes only the two rows $\mathbf{y}_i$ and $\mathbf{y}_j$ of the product $\mathbf{H}(t-1)\mathbf{W}^T(t)\mathbf{X}(t)$, which means that we need only the i^{th} and j^{th} rows of $\mathbf{H}(t-1)\mathbf{W}^T(t)$. We can write:

$$\mathbf{H}(t-1)\mathbf{W}^T(t) = \mathbf{B}(t-1) + \mathbf{H}(t-1)\mathbf{g}(t)\mathbf{e}'(t)^T \tag{16}$$

with $\mathbf{g}(t)$ and $\mathbf{e}'(t)$ are the FAPI rank one update vectors. Next, we calculate the rows $\mathbf{y}_i$ and $\mathbf{y}_j$ which will have $\mathcal{O}(nL)$ complexity with L the size of the data window ($L = 1/(1-\beta)$ in the case of the exponential window or the width of the truncated window). After that, one needs to optimize the function $J(\theta, \phi)$ and to update the outputs: i.e., $\mathbf{H}(t)$ by means of Eq. (11) and $\mathbf{B}(t)$ by combining Eq. (15) and Eq. (16). Note that the left multiplication by $\mathcal{S}_{ij}\mathcal{G}_{ij}$ resumes only in changing the i^{th} and j^{th} rows. The global complexity of the proposed algorithm is $5np + 2nL + \mathcal{O}(p^2)$ if we use the exponentially window version of FAPI or $8np+6nL+\mathcal{O}(p^2)$ if we use the truncated window version. The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 SGDS-FAPI

Require: $\mathbf{x}(t)$ data vector, parameters β and L, $\mathbf{W}(t-1)$ and $\mathbf{Z}(t-1)$ previous FAPI outputs. Previous indices (l, m) and outputs $\mathbf{H}(t-1)$, $\mathbf{B}(t-1)$.

Ensure: $\mathbf{W}(t)$, $\mathbf{Z}(t)$, indices (i, j), $\mathbf{H}(t)$ and $\mathbf{B}(t)$

1: Run FAPI : $\begin{cases} \vdots \\ \mathbf{W}(t) = \mathbf{W}(t-1) + \mathbf{e}'(t)\mathbf{g}(t)^T \end{cases}$
2: Update indices (i, j) using automatic selection strategy
3: Computing $\mathbf{y}_i$ and $\mathbf{y}_j$ using (16)
4: Find $(\hat{\theta}, \hat{\psi}) = \arg\min \quad J(\theta, \psi)$
5: $\mathbf{H}(t) = \mathcal{S}_{ij}\mathcal{G}_{ij}\mathbf{H}(t-1)$
6: $\mathbf{B}(t) = \mathcal{S}_{ij}\mathcal{G}_{ij}\mathbf{B}(t-1) + \mathbf{H}(t)\mathbf{g}(t)\mathbf{e}'(t)^T$

Besides their low computational complexity, the Jacobi-like techniques are also known for their ability to be easily parallelizable. This can be really useful in case where we consider more than one Shear-Givens rotation every time iteration. In this case, one need to be careful with the indices selection strategy used and repeat the steps 3-6 of Algorithm 1. Otherwise, one can just do this sequentially by repeating steps 2-6 of Algorithm 1. The latter has been tested in the sequel to assess the algorithm's performance in that case.

Remark: Note that other selection strategies for the rotation indices can be considered but are omitted due to space limitation. For example, at each iteration, one can select the indices (i, j) corresponding to row vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ of maximum ℓ_1 norms (i.e. the ones that deviate the most from the target sparsity objective).

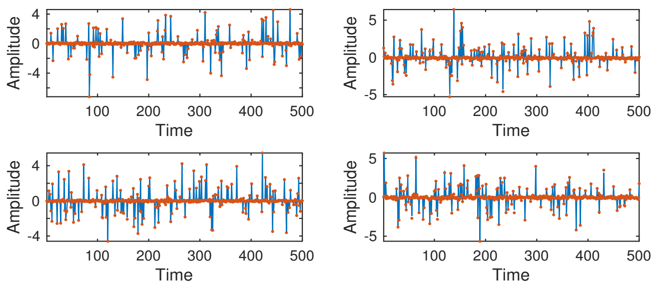

Fig. 1. Example of source signals (solid blue lines) and their separated versions (red points) for $d = 16$, $p = 4$ and $SNR = 10dB$.

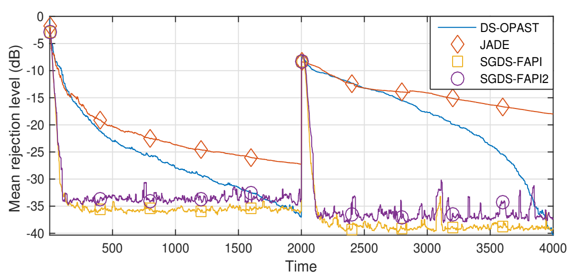

Fig. 2. Mean rejection level I_{pref} versus time for $d = 16$, $p = 4$ and $SNR = 20dB$.

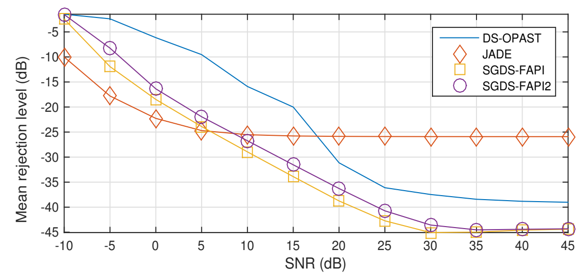

Fig. 3. Mean rejection level I_{pref} versus SNR for $d = 16$, $p = 4$ after 2000 iterations.

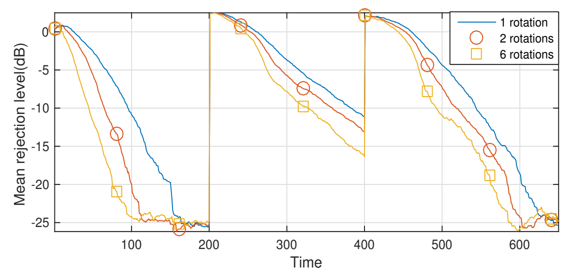

Fig. 4. Mean rejection level I_{pref} versus Time for $d = 20$, $p = 8$ and $SNR = 10db$.

4. SIMULATIONS RESULTS

In order to assess the performance of the proposed algorithm, we present here some numerical simulation results. The batch algorithm JADE [9] (re-applied at each time instant t to all samples from 1 to t), and the adaptive algorithm DS-OPAST [8] was used for comparison. We consider the data model presented in section 2 with the sparse signals generated according to the Bernoulli-Gaussian distribution (SPRANDN Matlab function). The performance index used is the mean rejection level [1], which is defined by $\mathcal{I}_{perf} \stackrel{def}{=} \sum_{p \neq q} \mathcal{I}_{pq}$ where $\mathcal{I}_{pq}$ measures the ratio of the power of the interference of the q^{th} source to the power of the p^{th} source signal. In our case, since the sources are generated with the same power, they are defined as $\mathcal{I}_{pq} = \mathrm{E}|(\hat{\mathbf{A}}^{\#}\mathbf{A})_{pq}|$ (where $\hat{\mathbf{A}}^{\#} = \mathbf{H}(t)\mathbf{W}^T(t)$).

We simulated 100 times the data with $p = 4$ sparse sources, $d = 16$ sensors. Figure 1 shows an example of source signals and their corresponding separated signals for a $SNR = 10dB$ (after adjusting the amplitude and put every output signal with its correspondent source signal to remove the inherent ambiguities of BSS). In order to show the adaptive separation capability of our algorithm, we change randomly the mixing matrix $\mathbf{A}$ after 2000 iterations. Fig.2 illustrates the improved performance of the SGDS-FAPI compared to DS-OPAST and JADE algorithms. SGDS-FAPI2 corresponds to truncated window version with $L = 50$ which explains its higher sensitivity to noise as compared to SGDS-FAPI. Fig.3 shows the results after 2000 iterations versus the SNR. It is clear that our algorithm reaches lower mean rejection levels than the other adaptive algorithms and even outperforms, in that context, the batch algorithm JADE for $SNR > 10dB$. In order to show the effect of using multiple rotations per iteration on the speed of convergence, we change the mixing matrix (randomly) two times: at time instants 200 and 400 with the parameters $d = 20$, $p = 8$ and $SNR = 10dB$. Fig.4 shows that the more rotations per iteration we consider, the faster is the convergence rate. The computational complexity should increase with such solution, unless we use a parallel scheme with an appropriate indices selection strategy.

5. CONCLUSION

The problem of blind adaptive sparse source separation has been studied in this paper. The over-determined instantaneous noisy mixture has been considered and we have proposed the two-step SGDS-FAPI algorithm. In the first step, we project the data on the signal subspace estimated by means of FAPI algorithm. The sparsity of the source signals is measured by an ℓ_1 criterion which is optimized by an adaptive method based on Shear and Givens rotations. In addition to the low computational cost, the proposed algorithm has shown improved performance as compared to the existing solutions.

6. REFERENCES

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation, Independent Component Analysis and Applications*, Academic Press (Elsevier), 2010.

[2] M. Congedo, C. Gouy-Pailler, and C. Jutten, "On the blind source separation of human electroencephalogram by approximate joint diagonalization of second order statistics.," *Clinical Neurophysiology*, vol. 119, no. 12, pp. 2677–2686, 2008.

[3] A. Ozerov and C. Fevotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[4] M. Valkama, M. Renfors, and V. Koivunen, "Advanced methods for i/q imbalance compensation in communication receivers," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2335–2344, 2001.

[5] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[6] A. Aïssa-El-Bey, K. Abed-Meraim, and Y. Grenier, "Blind audio source separation using sparsity based criterion for convolutive mixture case," in *Independent Component Analysis and Signal Separation (ICA)*, 2007, vol. 4666, pp. 317–324.

[7] J. Bobin, Jean-Luc Starck, Y. Moudden, and Jalal M. Fadili, "Blind Source Separation: the Sparsity Revolution," in *Advances in Imaging and Electron Physics*, ed. Peter Hawkes, Ed., vol. 152, pp. 221–306. Academic Press, Elsevier, 2008.

[8] N. Lassami, K. Abed-Meraim, and A. Aïssa-El-Bey, "Low cost subspace tracking algorithms for sparse systems," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1400–1404.

[9] J. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.

[10] S. A. W. Shah, K. Abed-Meraim, and T. Y. Al-Naffouri, "Blind source separation algorithms using hyperbolic and givens rotations for high-order qam constellations," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1802–1816, 2018.

[11] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on shear and givens rotations," *IEEE Transactions on Signal Processing*, vol. 62, no. 8, pp. 1913–1925, 2014.

[12] R. Badeau, G. Richard, B. David, and K. Abed-Meraim, "Approximated power iterations for fast subspace tracking," in *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, 2003, vol. 2, pp. 583–586.

[13] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2931–2941, 2005.

[14] C. Ramirez, V. Kreinovich, and M. Argaez, "Why l1 is a good approximation to l0: A geometric explanation," *Journal of Uncertain Systems*, vol. 7, pp. 203–207, 2013.