NON-UNITARY JOINT ZERO-BLOCK DIAGONALIZATION OF MATRICES USING A CONJUGATE GRADIENT ALGORITHM

O. Cherrak, H. Ghennioui, E.-H. Abarkan

Université Sidi Mohamed Ben Abdellah
FSTF, LSSC
Faculté des Sciences et Techniques de Fès,
Route Immouzzer, B.P. 2202, Fès-Maroc
{cherrakomar, ghennioui}@gmail.com

N. Thirion-Moreau, E. Moreau

Université Aix-Marseille, CNRS,
ENSAM, LSIS, UMR 7296,
13397 Marseille-France
Université de Toulon, CNRS, LSIS,
UMR 7296, 83957 La Garde-France
{thirion, moreau}@univ-tln.fr

ABSTRACT

This communication addresses a new problem which is the Non-Unitary Joint Zero-Block Diagonalization of a given set of complex matrices. This problem can occur in fields of applications such as blind separation of convolutive mixtures of sources and generalizes the non unitary Joint Zero-Diagonalization problem.
We present a new method based on the Conjugate Gradient algorithm. Our algorithm uses a numerical diagram of optimization which requires the calculation of the complex gradient matrix. The main advantages of the proposed method stem from the conjugate gradient properties: it is fast, stable and robust. Computer simulations are provided in order to illustrate the good behavior of the proposed method in different contexts. Two cases are studied: in the first scenario, a set of exactly zero-block-diagonal matrices are considered, then these matrices are progressively perturbed by an additive gaussian noise.

Index Terms— Joint zero-block diagonalization, matrix decompositions, conjugate gradient algorithm, linear convolutive mixtures.

1. INTRODUCTION

In the recent years, the problem of the joint decomposition of matrices (or tensors) sets has often arisen in the signal processing field, especially in blind source separation and array processing applications. One of the first considered problems was the Joint Diagonalization (JD) of a given matrix set under the unitary constraint, leading to the nowadays well-known JADE (Joint Approximate Diagonalization of Eigenmatrices) [1] and SOBI (Second Order Blind Identification) [2] algorithms. The following works have addressed either the problem of the JD of tensors [3][4] or the problem of JD of matrices but discarding the unitary constraint [5][6][7]. This first particular type of matrices decompositions is useful both in sources localization and direction finding problems and in blind sources separation of instantaneous mixtures.
A second type of matrices decompositions, namely the joint block-diagonalization (JBD), is encountered both in the wide-band sources localization in the presence of a correlated noise and in the blind separation of convolutive mixtures (or multi-dimensional deconvolution) problems. Several algorithms have been developed, under different assumptions about the considered matrix set (the matrices can be either positive definite or Hermitian) and about the block-diagonalizer (it is assumed unitary [8] or not [9][10] [11][12]). A third type of matrices decompositions, namely the joint zero-diagonalization (JZD), has proven to be useful in blind source separation, telecommunication and cryptography. The first suggested algorithms operated under the unitary constraint [13], since they were applied after a classical pre-whitening stage. But such a preliminary pre-whitening step establishes a bound with regard to the best reachable performances in the context of BSS that is the reason why the unitary constraint was soon discarded, leading to several other solutions [14][15][16]. In this communication, our purpose is to generalize the non unitary joint zero diagonalization approach suggested in [17][18] to the non-unitary joint zero-block diagonalization (JZBD) of several complex (not necessarily Hermitian) matrices (the zero-block-diagonalizer is not assumed unitary). It involves the choice of a well-chosen cost function and the calculation of quantities such as the complex gradient matrix. The main advantage of this approach remains its rather generic aspect since it encompasses the aforementioned JZD problem.

2. PROBLEM STATEMENT

2.1. Non-unitary joint zero-block diagonalization

The problem of the non-unitary joint zero-block diagonalization is stated in the following way. We consider a set $\mathcal{N}$ of N_m ($N_m \in \mathbb{N}^*$) square matrices $\mathbf{X}_i \in \mathbb{C}^{M\times M}$. For $i \in \{1,\ldots,N_m\}$, these matrices all admit the following

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

decomposition form: $\mathbf{X}_i = \mathbf{A}\mathbf{Z}_i\mathbf{A}^H$, where $(\cdot)^H$ stands for the transpose conjugate operator and the matrices $\mathbf{Z}_i = \begin{pmatrix} \mathbf{0}_{11} & \mathbf{Z}_{i,12} & \dots & \mathbf{Z}_{i,1r} \\ \mathbf{Z}_{i,21} & \mathbf{0}_{22} & \ddots & \mathbf{Z}_{i,2r} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{Z}_{i,r1} & \dots & \vdots & \mathbf{0}_{rr} \end{pmatrix}$, for all $i \in \{1, \dots, N_m\}$ are $(N \times N)$ zero-block diagonal matrices with r the number of considered blocks ($r \in \mathbb{N}^*$), $\mathbf{Z}_{i,kl}$, for all $k, l \in \{1, \dots, r\}$ are $n_k \times n_l$ matrices so that $n_1 + \dots + n_r = N$ where $\mathbf{0}_{kk}$ denotes the $n_k \times n_k$ square null matrix. $\mathbf{A}$ is a $M \times N$ ($M \geq N$) full rank matrix and the $N \times M$ matrix $\mathbf{B}$ is its pseudo-inverse (or generalized Moore-Penrose inverse). The set of the N_m square matrices $\mathbf{Z}_i \in \mathbb{C}^{N \times N}$ is denoted by $\mathcal{Z}$. The block sizes n_j for all $j = 1, \dots, r$ are assumed known.
The NU − JZBD problem consists of estimating $\mathbf{A}$ and the zero-block-diagonal matrices belonging to $\mathcal{Z}$ from only the matrix set $\mathcal{N}$. To tackle that problem, we propose, here, to consider the following cost function:

$$\mathcal{C}_{ZBD}(\mathbf{B}) = \sum_{i=1}^{N_m} \|\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\|_F^2, \qquad (1)$$

where the matrix operator $\mathsf{Bdiag}_{(\mathbf{n})}\{.\}$ is defined as follows:

$$\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{X}\} = \begin{pmatrix} \mathbf{X}_{11} & \mathbf{0}_{12} & \dots & \mathbf{0}_{1r} \\ \mathbf{0}_{21} & \mathbf{X}_{22} & \ddots & \mathbf{0}_{2r} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0}_{r1} & \mathbf{0}_{r2} & \dots & \mathbf{X}_{rr} \end{pmatrix}, \qquad (2)$$

where $\mathbf{X}$ is a $N \times N$ square matrix whose block components $\mathbf{X}_{ij}$ for all $i, j = 1, \dots, r$ are $n_i \times n_j$ matrices (and $n_1 + \dots + n_r = N$). Finally, we denote by $\mathbf{n} = (n_1, n_2, \dots, n_r)$. Using the cost function given Eq. (1) is rather advantageous since one single matrix is finally involved (direct estimation of the joint zero-block-diagonalizer matrix).

3. AN ALGORITHM TO SOLVE THE NU-JZBD

The cost function given in Eq. (1) has to be minimized to estimate the zero-block-diagonalizer matrix $\mathbf{B}$. To that aim, we suggest a new algorithm based on the conjugate gradient approach [19]. This iterative optimization method is well-known for its robustness and effectiveness. It has been widely used in various fields (*e.g.* neural networks [20], array processing [21] or blind sources separation [22]). But first, the complex gradient matrix $\mathbf{G} = \nabla_a \mathcal{C}_{ZBD}(\mathbf{B}) = 2\frac{\partial \mathcal{C}_{ZBD}(\mathbf{B})}{\partial \mathbf{B}^*}$ or its vectorization denoted by $\mathbf{g} = \mathbf{vec}\left(\nabla_a \mathcal{C}_{ZBD}(\mathbf{B})\right)$ has to be calculated. The operator $\mathbf{vec}(.)$ stacks the columns of a matrix into a vector. This calculation is performed in Appendix A. It is established that $\nabla_a \mathcal{C}_{ZBD}(\mathbf{B})$ is equal to:

$$2\sum_{i=1}^{N_m}\left[\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\mathbf{B}\mathbf{X}_i^H + \left(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right)^H \mathbf{B}\mathbf{X}_i\right].$$

3.1. Updating rule of the conjugate gradient algorithm

The matrix $\mathbf{B}$ is re-evaluated at each iteration m. Thus, it is denoted by $\mathbf{B}^{(m)}$ or by $\mathbf{b}^{(m)}$ when the vector $\mathbf{b}^{(m)} = \mathbf{vec}\left(\mathbf{B}^{(m)}\right)$ is considered instead of the matrix. The CG algorithm (with restarts) used to estimate $\mathbf{b}$ thus reads:
Step 1. Given $\mathbf{b}^{(0)}$, compute $\mathbf{g}^{(0)}$ and set $\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}$
Step 2. For $m = 0, 1, \dots, n-1$

$$\begin{cases} \mathbf{b}^{(m+1)} &= \mathbf{b}^{(m)} - \mu^{(m)}\mathbf{d}_{\mathbf{B}}^{(m)}, \\ \mathbf{d}_{\mathbf{B}}^{(m+1)} &= -\mathbf{g}^{(m+1)} + \beta^{(m)}\mathbf{d}_{\mathbf{B}}^{(m)}. \end{cases} \qquad (3)$$

Step 3. If $m = n-1$ replace $\mathbf{b}^{(0)}$ by $\mathbf{b}^{(n)}$ and go back to Step 1 (*i.e.* restart).
μ is a positive small factor called the step-size and $\mathbf{d}_{\mathbf{B}}$ is the direction of search. We show in Section 3.2 how the optimal step-size μ_{opt} can be calculated at each iteration. In exact line search methods, several expressions of β have been suggested among which are the Fletcher-Reeves (β_{FR}), the Polak-Ribière (β_{PR}) [23][24] and the Dai-Yuan (β_{DY}) formula [25]:

$$\beta_{\mathsf{FR}}^{(m+1)} = \frac{(\mathbf{g}^{(m+1)})^H\mathbf{g}^{(m+1)}}{(\mathbf{g}^{(m)})^H\mathbf{g}^{(m)}}, \qquad (4)$$

$$\beta_{\mathsf{PR}}^{(m+1)} = \frac{(\mathbf{g}^{(m+1)}-\mathbf{g}^{(m)})^H\mathbf{g}^{(m+1)}}{(\mathbf{g}^{(m)})^H\mathbf{g}^{(m)}}, \qquad (5)$$

$$\beta_{\mathsf{DY}}^{(m+1)} = \frac{(\mathbf{g}^{(m+1)})^H\mathbf{g}^{(m+1)}}{(\mathbf{d}_{\mathbf{B}}^{(m)})^H(\mathbf{g}^{(m+1)}-\mathbf{g}^{(m)})}. \qquad (6)$$

The restarting aspect is important for the convergence of the algorithm since in general one cannot guarantee that the directions $\mathbf{d}^{(k)}$ are effectively descent directions. But since a pure steepest descent step is taken at each restart by setting $\mathbf{d}_{\mathbf{B}}^{(m)} = -\mathbf{g}^{(m)}$ the local convergence is assured.

3.2. Exact Line Search: seek of the optimal step-size

The optimal step size, say μ_{opt}, is used to decrease the total number of iterations needed to reach convergence. It minimizes the polynomial $\mathcal{C}_{ZBD}(\mathbf{P})$ where $\mathbf{P} = \mathbf{B}^{(m)} - \mu\mathbf{D}_{\mathbf{B}}^{(m)}$ and $\mathbf{d}_{\mathbf{B}}^{(m)} = \mathbf{vec}\left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)$. In our case, this quantity is a 4th order polynomial, and its derivative a 3rd order polynomial. They are respectively given by:

$$\mathcal{C}_{ZBD}(\mathbf{B} - \mu\mathbf{D}_{\mathbf{B}}) = a_0 + a_1\mu + a_2\mu^2 + a_3\mu^3 + a_4\mu^4, \qquad (7)$$

$$\frac{\partial \mathcal{C}_{ZBD}(\mathbf{B}-\mu\mathbf{D}_{\mathbf{B}})}{\partial \mu} = 4a_4\mu^3 + 3a_3\mu^2 + 2a_2\mu + a_1, \qquad (8)$$

where the five coefficients a_i for $i = 0, \dots, 4$ are equal to (see Appendix B):

$$a_0 = \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_3^H\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_3\}\right\},$$

$$a_1 = -\sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_1\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_3\} + \mathbf{K}_3^H\mathbf{K}_2\right\},$$

$$a_2 = \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_0^H\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_3\} + \mathbf{K}_3^H\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_0\} + \mathbf{K}_1\mathbf{K}_2\right\},$$

$$a_3 = -\sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_0^H\mathbf{K}_2 + \mathbf{K}_1\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_0\}\right\},$$

$$a_4 = \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_0^H\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_0\}\right\},$$

where:
$\mathbf{K}_0 = \mathbf{D}_{\mathbf{B}}^{(m)} \mathbf{X}_i (\mathbf{D}_{\mathbf{B}}^{(m)})^H$,
$\mathbf{K}_1 = \mathbf{D}_{\mathbf{B}}^{(m)} \mathbf{X}_i^H \mathbf{B}^H + \mathbf{B} \mathbf{X}_i^H (\mathbf{D}_{\mathbf{B}}^{(m)})^H$,
$\mathbf{K}_2 = \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i (\mathbf{D}_{\mathbf{B}}^{(m)})^H + \mathbf{D}_{\mathbf{B}}^{(m)} \mathbf{X}_i \mathbf{B}^H\}$,
$\mathbf{K}_3 = \mathbf{B}\mathbf{X}_i\mathbf{B}^H$.
The optimal step-size μ_{opt} corresponds to the root of (8) which minimizes (7).

3.3. Algorithm

The non-unitary JZBD algorithm (denoted by $\mathsf{JZBD}_{\mathsf{CG}}$) is summarized below:

Data: N_m square matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_m}$, stopping criterion ϵ, step-size μ, max. number of iterations M_{max}
Result: Estimation of joint zero block diagonalizer $\mathbf{B}$
initialize: $\mathbf{B}^{(0)}$; $\mathbf{D}^{(0)}$; $m = 0$;
repeat
  if $m \mod M_0 = 0$ **then**
    restart
  else
    Calculate optimal step-size $\mu_{\mathsf{opt}}^{(m)}$
    Compute gradient matrix $\nabla_a \mathcal{C}_{ZBD}(\mathbf{B}^{(m+1)})$
    Compute matrix $\mathbf{B}^{(m+1)}$
    Compute $\beta_{\mathsf{PR}}^{(m)}$
    Compute the search direction $\mathbf{D}_{\mathbf{B}}^{(m+1)}$
    $m = m + 1$;
  end
until *((*$|\mathbf{B}^{(m+1)} - \mathbf{B}^{(m)}| \leq \epsilon$*) or (*$m \geq M_{max}$*))*;

4. COMPUTER SIMULATIONS

We present simulations to illustrate the effectiveness of the suggested algorithm. We consider a set $\mathcal{Z}$ of $N_m = 5$ (resp. 20 and 100) matrices, randomly chosen according to a Gaussian law of mean 0 and variance 1. First, these matrices are considered as exactly zero block-diagonal, then a random matrix of noise is added (drawn from a gaussian law of mean 0 and variance σ_b^2). The signal to noise ratio is defined as $\mathsf{SNR} = 10\log(\frac{1}{\sigma_b^2})$. To measure the quality of the estimation, the ensuing error index (which is an extension of the one introduced in [4]) is used:

$$I_{conv}(\mathbf{G}) = \frac{1}{r(r-1)} \left[\sum_{i=1}^{r} \left(\sum_{j=1}^{r} \frac{\|(\mathbf{G})_{i,j}\|_F^2}{\max\limits_{\ell} \|(\mathbf{G})_{i,\ell}\|_F^2} - 1 \right) + \sum_{j=1}^{r} \left(\sum_{i=1}^{r} \frac{\|(\mathbf{G})_{i,j}\|_F^2}{\max\limits_{\ell} \|(\mathbf{G})_{\ell,j}\|_F^2} - 1 \right) \right],$$

where $(\mathbf{G})_{i,j}$ for all $i, j \in \{1, \ldots, r\}$ is the (i,j)-th block matrix of $\mathbf{G} = \hat{\mathbf{B}}\mathbf{A}$. The best results are obtained when the error index $I_{conv}(\cdot)$ is close to 0 in a linear scale ($-\infty$ in a logarithmic scale). All the displayed results have been averaged over 100 Monte-Carlo trials. For all simulations we have considered a size r of blocks equal to 4.

First, we plot on the Fig.1 the evolution of the performance of the suggested algorithm versus the iterations for different sizes of the matrix sets ($N_m = 5$, 20 and 100). In this example, we consider the noiseless case where $\mathsf{SNR} = 100$ dB. Whereas, on the Fig. 2, the same study is performed in a noisy context ($\mathsf{SNR} = 25$ dB). In both cases, the algorithm is initialized thanks to the generalized eigenvalue decomposition [10].

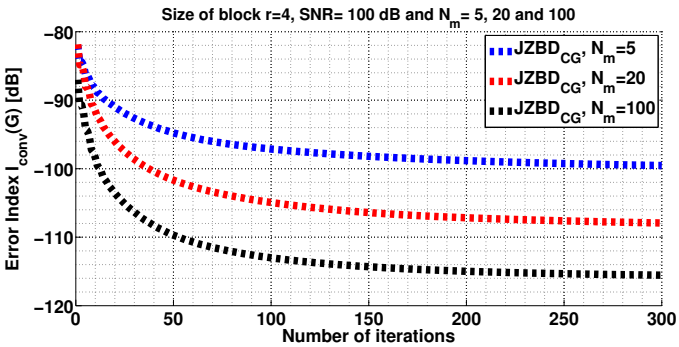

Fig. 1. Evolution of the error index $I_{conv}(\mathbf{G})$ versus the number of iterations in the noiseless case ($\mathsf{SNR} = 100$ dB) and for N_m =5, 20 and 100 matrices. We consider the case of $r = 4$ blocks.

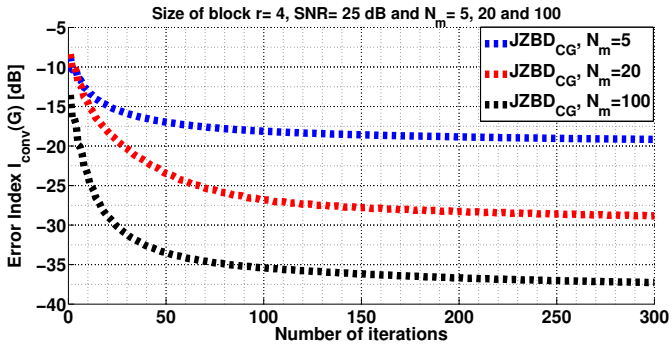

Fig. 2. Evolution of the error index $I_{conv}(\mathbf{G})$ versus the number of iterations in the noisy case ($\mathsf{SNR} = 25$ dB) and for N_m =5, 20 and 100 matrices. We consider the case of $r = 4$ blocks.

In the noiseless case, we can observe the good stability and convergence of the $\mathsf{JZBD}_{\mathsf{CG}}$ algorithm for $N_m = 5$ (resp. 20 and 100), since it reaches -100 dB (resp. -108 dB and -115 dB). In a noisy context, we observe the same kind of behavior, except that the performance are deteriorated and bounded (-37 dB instead of -115 dB).
Finally, on Fig.3, we emphasize the influence of the SNR. We display the evolution of the error index versus the number of matrices N_m. Different values of the SNR are considered (25 dB, 40 dB and 100 dB). After that, we show the influence of the size N_m of the matrix set to be joint zero block-diagonalized. We display the evolution of the error index versus the SNR for different sizes of the matrix sets ($N_m = 5$, 20 and 100). These charts illustrate the behavior of the proposed algorithm *i.e.* an increase of the performance when bigger subsets of matrices are considered or when the SNR is

improved.

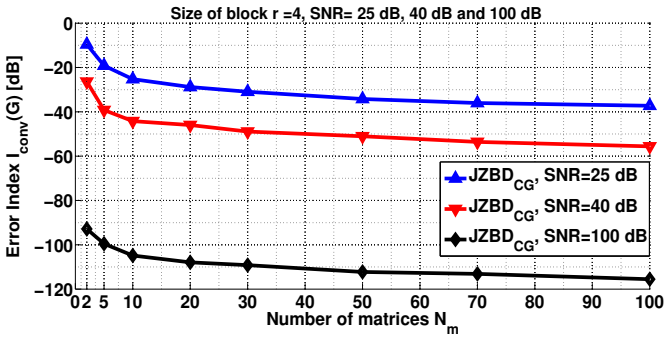

Fig. 3. Evolution of the error index $I_{conv}(\mathbf{G})$ versus the number, N_m, of matrices that are used for SNR $= 25$ dB, 40 dB and 100 dB. We consider the case of $r = 4$ blocks.

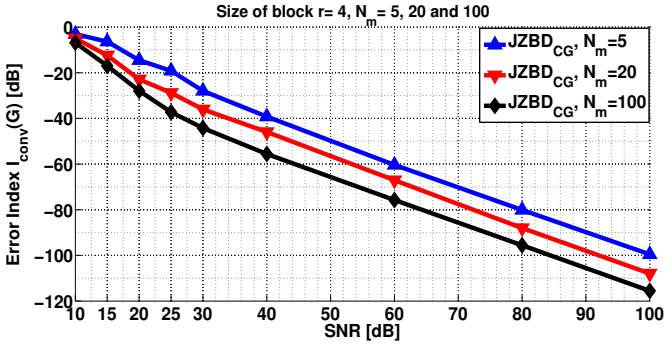

Fig. 4. Evolution of the error index $I_{conv}(\mathbf{G})$ versus the SNR for different sizes of the matrix sets $N_m = 5$, 20 and 100. We consider the case of $r = 4$ blocks.

5. CONCLUSION

In this communication, we have suggested a new NU $-$ JZBD algorithm called JZBD$_{\mathsf{CG}}$. It relies on a conjugate gradient optimization scheme which requires the calculation of the complex gradient matrix. The main advantage of the JZBD$_{\mathsf{CG}}$ algorithm is its more general aspect as well as the fact that it leads to rather good performances even in difficult situations (noise and/or few matrices to be joint zero block-diagonalized). Further works will consist of showing the interest of this algorithm in a true blind multi-dimensional deconvolution context (blind separation of convolutive mixtures of sources).

APPENDIX

Considering three $(M \times M)$ square matrices $\mathbf{D}_1$, $\mathbf{D}_2$ and $\mathbf{D}_3$ and two rectangular matrices $\mathbf{D}_4$ $(M \times N)$ and $\mathbf{D}_5$ $(N \times M)$ and a square $N \times N$ matrix $\mathbf{D}_6$, let $\mathsf{tr}\{\cdot\}$, $\mathsf{d}\{\cdot\}$, $\mathsf{vec}\{\cdot\}$, $\mathsf{Bdiag}_{(\mathbf{n})}\{\cdot\}$ respectively denote the trace operator, the differential operator, the vec-operator, the block-diagonal operator defined in Eq. (2) and $\mathbf{T}_{\mathrm{BDiag}} = \mathsf{diag}\{\mathbf{vec}(\mathsf{BDiag}\{\mathbf{1}_N\})\}$, $\mathbf{1}_N$ is the $N \times N$ matrix whose components are all ones. Our developments are based on the ensuing properties:

$\mathbf{P}_1$. $\|\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1\}\|_F^2 = \mathsf{tr}\{(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1\})^H \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1\}\}$ $= \mathsf{tr}\{\mathbf{D}_1^H \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1\}\}$.
$\mathbf{P}_2$. $\mathsf{tr}\{\mathbf{D}_1\} = \mathsf{tr}\{\mathbf{D}_1^T\}$.
$\mathbf{P}_3$. $\mathsf{tr}\{\mathbf{D}_1 + \mathbf{D}_2\} = \mathsf{tr}\{\mathbf{D}_1\} + \mathsf{tr}\{\mathbf{D}_2\}$.
$\mathbf{P}_4$. $\mathsf{tr}\{\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3\} = \mathsf{tr}\{\mathbf{D}_3\mathbf{D}_1\mathbf{D}_2\} = \mathsf{tr}\{\mathbf{D}_2\mathbf{D}_3\mathbf{D}_1\} \Rightarrow$ $\mathsf{tr}\{\mathbf{D}_1\mathbf{D}_2\} = \mathsf{tr}\{\mathbf{D}_2\mathbf{D}_1\}$.
$\mathbf{P}'_4$. $\mathsf{tr}\{\mathbf{D}_4\mathbf{D}_5\} = \mathsf{tr}\{\mathbf{D}_5\mathbf{D}_4\}$.
$\mathbf{P}_5$. $\mathsf{tr}\{\mathbf{D}_1^H\mathbf{D}_2\} = (\mathsf{vec}\{\mathbf{D}_1\})^H \mathsf{vec}\{\mathbf{D}_2\}$.
$\mathbf{P}_6$. $\mathsf{vec}\left\{\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_6\}\right\} = \mathbf{T}_{\mathrm{BDiag}}\mathsf{vec}\{\mathbf{D}_6\}$.
$\mathbf{P}_7$. $\mathsf{d}\{\mathbf{D}_1^H\} = (\mathsf{d}\{\mathbf{D}_1\})^H$.
$\mathbf{P}_8$. $\mathsf{d}\{\mathbf{D}_1^*\} = (\mathsf{d}\{\mathbf{D}_1\})^*$.
$\mathbf{P}_9$. $\mathsf{d}\{\mathbf{D}_1\mathbf{D}_2\} = \mathsf{d}\{\mathbf{D}_1\}\mathbf{D}_2 + \mathbf{D}_1\mathsf{d}\{\mathbf{D}_2\}$.
$\mathbf{P}_{10}$. $\mathsf{d}\{\mathbf{D}_1 + \mathbf{D}_2\} = \mathsf{d}\{\mathbf{D}_1\} + \mathsf{d}\{\mathbf{D}_2\}$.
$\mathbf{P}_{11}$. $\mathsf{d}\{\mathsf{tr}\{\mathbf{D}_1\}\} = \mathsf{tr}\{\mathsf{d}\{\mathbf{D}_1\}\}$.
$\mathbf{P}_{12}$. $\mathsf{d}\{\mathsf{vec}\{\mathbf{D}_1\}\} = \mathsf{vec}\{\mathsf{d}\{\mathbf{D}_1\}\}$.
$\mathbf{P}_{13}$. if $f(\mathbf{Z}, \mathbf{Z}^*) = \mathsf{tr}\{\mathbf{D}_1^T\mathbf{Z}+\mathbf{Z}^H\mathbf{D}_2\}$ then $\mathsf{d}\{f(\mathbf{Z}, \mathbf{Z}^*)\} =$ $\mathsf{tr}\left\{\mathbf{D}_1^T d\mathbf{Z} + \mathbf{D}_2^T d\mathbf{Z}^*\right\} \Rightarrow \frac{\partial f}{\partial \mathbf{Z}} = \mathbf{D}_1$ and $\frac{\partial f}{\partial \mathbf{Z}^*} = \mathbf{D}_2$.
$\mathbf{P}_{14}$. $(\mathbf{D}_1\mathbf{D}_2)^H = \mathbf{D}_2^H\mathbf{D}_1^H$.
$\mathbf{P}_{15}$. $(\mathbf{D}_1 + \mathbf{D}_2)^H = (\mathbf{D}_1^H + \mathbf{D}_2^H)$.
$\mathbf{P}_{16}$. $\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1 + \mathbf{D}_2\} = \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_1\} + \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{D}_2\}$.

A. Calculation of the complex gradient matrix of the cost function $\mathcal{C}_{ZBD}(\mathbf{B})$

Using the property $\mathbf{P}_1$, $\mathbf{P}_3$, $\mathbf{P}_9$, $\mathbf{P}_{10}$ and $\mathbf{P}_{11}$, the differential of cost function is rewritten as:

$$\begin{aligned}\mathsf{d}\{\mathcal{C}_{ZBD}(\mathbf{B})\} &= \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathsf{d}\{(\mathbf{B}\mathbf{X}_i\mathbf{B}^H)^H \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\}\right\}, \\ &= \mathcal{F}(\mathbf{B}) + \mathcal{G}(\mathbf{B}), \end{aligned} \quad (9)$$

where, $\mathcal{F}(\mathbf{B}) = \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathsf{d}\left\{(\mathbf{B}\mathbf{X}_i\mathbf{B}^H)^H\right\} \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right\}$, and $\mathcal{G}(\mathbf{B}) = \sum_{i=1}^{N_m} \mathsf{tr}\left\{(\mathbf{B}\mathbf{X}_i\mathbf{B}^H)^H \mathsf{d}\left\{\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right\}\right\}$. After using the properties $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}'_4$, $\mathbf{P}_7$, $\mathbf{P}_9$, $\mathbf{P}_8$, $\mathbf{P}_{12}$ $\mathcal{F}(\mathbf{B})$ and $\mathcal{G}(\mathbf{B})$ in Eq. (9) and using the property $\mathbf{P}_{13}$ we obtain $\frac{\partial \mathcal{C}_{ZBD}(\mathbf{B})}{\partial \mathbf{B}}$ and $\frac{\partial \mathcal{C}_{ZBD}(\mathbf{B})}{\partial \mathbf{B}^*}$ respectively equal as follows:

$$\sum_{i=1}^{N_m} \left(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right)^T \mathbf{B}^*\mathbf{X}_i^* + \sum_{i=1}^{N_m} \left(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right)^* \mathbf{B}^*\mathbf{X}_i^T,$$

$$\sum_{i=1}^{N_m} \left(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right) \mathbf{B}\mathbf{X}_i^H + \sum_{i=1}^{N_m} \left(\mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i\mathbf{B}^H\}\right)^H \mathbf{B}\mathbf{X}_i.$$

It finally leads to the result stated in Section 3.

B. Coefficients of the 4th-degree polynomial

Using the properties $\mathbf{P}_{14}$ and $\mathbf{P}_{15}$, the cost function can be expressed as:

$$\begin{aligned}\mathcal{C}_{ZBD}(\mathbf{B} - \mu\mathbf{D}_{\mathbf{B}}^{(m)}) = \sum_{i=1}^{N_m} \Big|\Big| \mathsf{Bdiag}_{(\mathbf{n})} \Big\{ \mu^2 \left(\mathbf{D}_{\mathbf{B}}^{(m)}\mathbf{X}_i \left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)^H\right) \\ - \mu\left(\mathbf{B}\mathbf{X}_i \left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)^H + \mathbf{D}_{\mathbf{B}}^{(m)}\mathbf{X}_i\mathbf{B}^H\right) + \mathbf{B}\mathbf{X}_i\mathbf{B}^H \Big\} \Big|\Big|_F^2. \end{aligned} \quad (10)$$

From the properties $\mathbf{P}_1$, $\mathbf{P}_{14}$,$\mathbf{P}_{15}$ and $\mathbf{P}_{16}$ when introducing the four following matrices $\mathbf{K}_0, \mathbf{K}_1, \mathbf{K}_2$ and $\mathbf{K}_3$:

$$\mathbf{K}_0 = \mathbf{D}_{\mathbf{B}}^{(m)}\mathbf{X}_i \left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)^H, \quad (11)$$

$$\mathbf{K}_1 = \mathbf{D}_{\mathbf{B}}^{(m)} \mathbf{X}_i^H \mathbf{B}^H + \mathbf{B}\mathbf{X}_i^H \left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)^H, \quad (12)$$

$$\mathbf{K}_2 = \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{B}\mathbf{X}_i \left(\mathbf{D}_{\mathbf{B}}^{(m)}\right)^H + \mathbf{D}_{\mathbf{B}}^{(m)} \mathbf{X}_i \mathbf{B}^H\}, \quad (13)$$

$$\mathbf{K}_3 = \mathbf{B}\mathbf{X}_i\mathbf{B}^H. \quad (14)$$

We finally find that:

$$\begin{aligned}
\mathcal{C}_{BD}\left(\mathbf{B} - \mu \mathbf{D}_{\mathbf{B}}^{(m)}\right) &= \sum_{i=1}^{N_m} \mathsf{tr}\left\{(\mathbf{K_3})^H \, \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K_3}\}\right\} \\
&- \mu \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_1 \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K_3}\} + (\mathbf{K_3})^H \mathbf{K_2}\right\} \\
&+ \mu^2 \sum_{i=1}^{N_m} \mathsf{tr}\left\{\mathbf{K}_0^H \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K_3}\} + (\mathbf{K_3})^H \, \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K_0}\} + \mathbf{K_1}\mathbf{K_2}\right\} \\
&- \mu^3 \sum_{i=1}^{N_m} \mathsf{tr}\left\{(\mathbf{K}_0)^H \mathbf{K}_2 + \mathbf{K}_1 \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_0\}\right\} \\
&+ \mu^4 \sum_{i=1}^{N_m} \mathsf{tr}\left\{(\mathbf{K}_0)^H \, \mathsf{Bdiag}_{(\mathbf{n})}\{\mathbf{K}_0\}\right\} \quad (15)
\end{aligned}$$

It finally leads to the results stated in Section 3.2.

6. REFERENCES

[1] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," *IEEE Proc.-F*, vol. 40, no. 6, pp. 362–370, 1993.

[2] A. Belouchrani, K. Abed-Meraïm, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. Signal Proc.*, vol. 45, pp. 434–444, Feb. 1997.

[3] P. Comon, "Independant component analysis, a new concept ?," *Signal Proc.*, vol. 36, pp. 287–314, 1994.

[4] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE Trans. Signal Proc.*, vol. 49, no. 3, pp. 530–541, March 2001.

[5] A. Yeredor, A. Ziehe, and K.-R. Müller, "Approximate joint diagonalization using natural gradient approach," in *Lecture Notes in Computer Science, in Proceedings ICA*, Granada, Espagne, May 2004, pp. 89–96.

[6] A. Souloumiac, "Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations," *IEEE Trans. Signal Proc.*, vol. 57, no. 6, pp. 2222–2231, June 2009.

[7] G. Chabriel and J. Barrère, "A direct algorithm for nonorthogonal approximate joint diagonalization," *IEEE Trans. Signal Proc.*, vol. 60, no. 1, pp. 39–47, 2012.

[8] L. De Lathauwer, C. Févotte, B. De Moor, and J. Vandewalle, "Jacobi algorithm for block diagonalization in blind identification," in *Proc. Symp. Inform. Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 2002, pp. 155–162.

[9] H. Ghennioui, N. Thirion-Moreau, E. Moreau, and D. Aboutajdine, "Gradient-based joint block diagonalization algorithms: Application to blind separation of fir convolutive mixtures," *Signal Proc.*, vol. 90, no. 6, pp. 1836–1849, June 2010.

[10] D. Nion, "A tensor framework for non-unitary joint block diagonalization," *IEEE Trans. Signal Proc.*, vol. 59, no. 10, pp. 4585–4594, Oct. 2011.

[11] P. Tichavsky and Z. Koldovsky, "Algorithms for nonorthogonal approximate joint block-diagonalization," in *Proceedings EUSIPCO*, Bucharest, Romania, August 2012, pp. 2094–2098.

[12] O. Cherrak, H. Ghennioui, E.-H. Abarkan, and N. Thirion-Moreau, "Non-unitary joint block diagonalization of matrices using a levenberg-marquardt algorithm," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Sept 2013, pp. 1–5.

[13] A. Belouchrani, K. Abed-Meraïm, M. G. Amin, and A. Zoubir, "Joint anti-diagonalization for blind source separation," in *Proc. IEEE ICASSP*, Salt Lake City, UT, May 2001, pp. 2196–2199.

[14] E. M. Fadaili, N. Thirion-Moreau, and E. Moreau, "Non-orthogonal joint diagonalization/zero-diagonalization for source separation based on time-frequency distributions," *IEEE Trans. Signal Proc.*, vol. 55, no. 5, May 2007.

[15] G. Chabriel, J. Barrère, N. Thirion-Moreau, and E. Moreau, "Algebraic joint zero-diagonalization and blind source separation," *IEEE Trans. Signal Proc.*, vol. 56, no. 3, pp. 980–989, March 2008.

[16] W. T. Zhang and S. T. Lou, "A low complexity iterative algorithm for joint zero-diagonalization," *IEEE Signal Proc. Lett.*, vol. 19, no. 2, pp. 115–118, January 2012.

[17] E. M. Fadaili, N. Thirion-Moreau, and E. Moreau, "Algorithme de zéro-diagonalisation conjointe pour la sparation de sources dterministes," in *20me Colloque GRETSI*, Louvain-La-Neuve, Belgique, Septembre 2005.

[18] E. M. Fadaili, N. Thirion-Moreau, and E. Moreau, "Combined non-orthogonal joint zero-diagonalization and joint diagonalization for source separation," in *IEEE Workshop on Statistical Signal Processing (SSP'2005)*, Bordeaux, France, Juillet 2005.

[19] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[20] N. Gong, W. Shao, and H. Xu, "The conjugate gradient method with neural network control," in *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, Nov 2010, pp. 82–84.

[21] A. Boonpoonga, S. Buritramart, P. Sirisuk, M. Krairiksh, and T.K. Sarkar, "Adaptive array processing unit based on direct data domain least square approach using conjugate gradient method," in *Microwave Conference, 2007. APMC 2007. Asia-Pacific*, Dec 2007, pp. 1–4.

[22] L. Zhang, "Conjugate gradient approach to blind separation of temporally correlated signals," in *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, June 2004, vol. 2, pp. 1008–1012.

[23] E. Polak, *Optimization algorithms and consistent approximations*, Springer, 1997.

[24] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific J. Optimiz*, vol. 2, pp. 35–58, 2006.

[25] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM J. Optim*, vol. 10, pp. 177–182, 1999.