A “Sequentially Drilled” Joint Congruence (SeD-JoCo) Transformation With Applications in Blind Source Separation and Multiuser MIMO Systems

Arie Yeredor, *Senior Member, IEEE*, Bin Song, *Student Member, IEEE*, Florian Roemer, *Student Member, IEEE*, and Martin Haardt, *Senior Member, IEEE*

***Abstract*—We consider a particular form of the classical approximate joint diagonalization (AJD) problem, which we call a “sequentially drilled” joint congruence (SeDJoCo) transformation. The problem consists of a set of symmetric real-valued (or Hermitian-symmetric complex-valued) target-matrices. The number of matrices in the set equals their dimension, and the joint diagonality criterion requires that in each transformed (“diagonalized”) target-matrix, all off-diagonal elements on one specific row and column (corresponding to the matrix-index in the set) be exactly zeros, yet does not care about the other (diagonal or off-diagonal) elements. The motivation for this form arises in (at least) two different contexts: maximum likelihood blind (or semiblind) source separation and coordinated beamforming for multiple-input multiple-output (MIMO) broadcast channels. We prove that SeDJoCo always has a solution when the target-matrices are positive-definite . We also propose two possible iterative solution algorithms, based on defining and optimizing two different criteria functions, using Newton’s method for the first function and successive Jacobi-like transformations for the second. The algorithms’ convergence behavior and the attainable performance in the two contexts above are demonstrated in simulation experiments.**

***Index Terms*—Approximate joint diagonalization, blind source separation, independent component analysis, coordinated beamforming, multi-user MIMO, STJOCO, HEAD.**

I. Introduction

The general framework of approximate joint diagonalization (AJD) considers a set of K (typically more than two) square, symmetric, real-valued $N \times N$ matrices denoted $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_K$ (often termed the “target-matrices”). The goal in AJD is to find a single matrix $\boldsymbol{B}$ (or its inverse $\boldsymbol{A}$, see below) which best “jointly diagonalizes” the target-matrices in some sense. AJD is closely related to the problem of blind source separation (BSS), in which the diagonalizing $\boldsymbol{B}$ serves as an estimate of the demixing matrix, which is subsequently used for recovering the sources from their observed mixtures. The matrix $\boldsymbol{A}$ serves, in turn, as an estimate of the unknown mixing matrix.

Quite a few approaches to the AJD problem have been suggested in the past two decades (e.g., [1]–[8]), mainly differing in the proposed criteria for measuring the extent of attained joint diagonalization. These can be generally divided into “direct” criteria, looking for $\boldsymbol{B}$ which makes all $\boldsymbol{B}\boldsymbol{Q}_k\boldsymbol{B}^T$ (where $(\cdot)^T$ denotes the transpose) “as diagonal as possible,” and “indirect” criteria, looking for $\boldsymbol{A}$ (and K diagonal matrices $\boldsymbol{D}_k$) such that all $\boldsymbol{Q}_k$ are “best fitted” with $\boldsymbol{A}\boldsymbol{D}_k\boldsymbol{A}^T$. When $\boldsymbol{B}$ (or $\boldsymbol{A}$) is restricted (such as in [1]) to be orthonormal, the problem is commonly referred to as *orthogonal AJD*, otherwise it is *nonorthogonal AJD*.

In this paper, we consider a somewhat less familiar, very particular form of (nonorthogonal) AJD, which can be viewed either as a specially structured joint congruence (STJOCO [9]) relation, or as “hybrid” exact-approximate joint diagonalization (HEAD, [10]), satisfied both in its “direct” and in its “indirect” formulation. For reasons that are explained below, we shall, from now on, refer to this particular form as a “sequentially drilled” joint congruence (SeDJoCo) transformation.

Unlike the problem of general AJD, the basic form of SeDJoCo considers exactly $K = N$ target-matrices $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_N$ (namely, the number of matrices equals their dimension), and seeks a matrix $\boldsymbol{B}$, such that the nth row and nth column of the transformed nth matrix $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^T$ would be all-zeros, except for the diagonal (n, n)th element. For each matrix $\boldsymbol{Q}_n$, this structure resembles a square that has been “drilled” along the nth row and column (considering the elements which have been zeroed-out as “empty”). Since the index of the “drilled” row and column progresses sequentially with the matrix-index, we call this congruence transformation “sequentially drilled”—hence the term SeDJoCo , see, for example, Fig. 1, which depicts the case of $N = 5$. We shall consider both the real-valued formulation, in which all the N target-matrices, as well as $\boldsymbol{B}$, are real-valued; and the complex-valued formulation, where all $\boldsymbol{Q}_n$, as well as $\boldsymbol{B}$, may be complex-valued (in which case the desired congruence relation takes the form $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$, with $(\cdot)^H$ denoting the conjugate-transpose).

Manuscript received March 03, 2011; revised October 11, 2011; accepted February 27, 2012. Date of publication March 13, 2012; date of current version May 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Patrick Flandrin.

A. Yeredor is with the School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel (e-mail: arie@eng.tau.ac.il).

B. Song, F. Roemer, and M. Haardt are with the Communications Research Laboratory, Ilmenau University of Technology, D-98684 Ilmenau, Germany (e-mail: bin.song@tu-ilmenau.de; florian.roemer@tu-ilmenau.de; martin.haardt@tu-ilmenau.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2190728

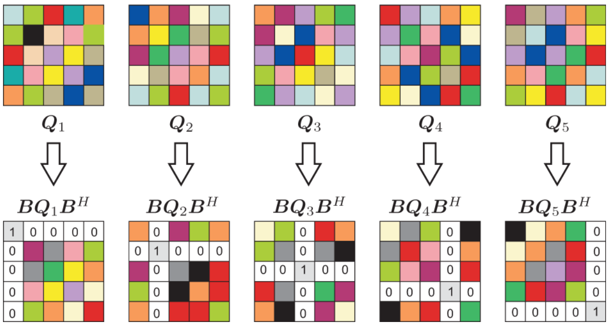


Fig. 1. SeDJoCo transformation of five target-matrices.

The motivation for seeking this kind of transformations stems from (at least) two different origins. One is encountered in the context of maximum likelihood (ML) blind (or semiblind) source separation; another is encountered in the context of coordinated beamforming (CBF) for multiple-input multiple-output (MIMO) broadcast channels. We shall elaborate on these two contexts in the sequel.

To the best of our knowledge, the first formulation of the associated set of equations was derived by Pham and Garat in [11], in the context of quasi-ML (QML) blind separation of stationary sources. Although the set of equations in [11] was equivalent to SeDJoCo, there was no explicit association of these equations in [11] with a form of joint diagonalization. A Newton-based iterative solution was proposed, but involved an approximation which assumed that the sources are nearly separated. Later on, Dégerine and Zaïdi derived in [12] a similar set of equations in the context of ML blind separation of Gaussian autoregressive (AR) sources and proposed a solution approach termed "iterative relaxation." Both solutions (in [11] and in [12]) only addressed the real-valued version.

A unified view of the set of equations, presented as a special form of joint diagonalization (termed "HEAD"), was proposed by Yeredor in [10] (in its real-valued form). It was further shown by Yeredor in [13], that ML separation of general (not necessarily stationary) Gaussian sources requires the solution of this set of equations (with various forms of target-matrices).

In independent, parallel work [9], Song *et al.* considered the same set of equations in a different context—of CBF in MIMO channels. The problem was termed "STJOCO" in [9], and a different iterative solution was proposed.

The intriguing occurrence of similar forms of this set of equations in such different contexts (with different interpretations for the target-matrices $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$ and for the transformation matrix $\boldsymbol{B}$) has motivated us to present an integrated, self-contained, comprehensive view of the problem (newly termed "SeDJoCo" in this work), along with some theoretical analysis, proposed solutions, performance study and application examples, both for the real-valued and complex-valued cases.

Our main contributions in this work are as follows:

- presentation of different equivalent formulations of the problem and their association with the joint diagonalization problem;
- a proof of existence of a solution for positive-definite target-matrices, both in the real-valued and complex-valued cases;
- two iterative solution algorithms, both based on formulating the problem as different optimization problems and taking different approaches in the maximization or minimization of the associated cost-functions. Both approaches are provided in both a real-valued and a complex-valued version:
  - —An approach based on Newton's method: Normally, the application of Newton's method would require the inversion of an $N^2 \times N^2$ Hessian matrix in each iteration, which might be computationally expensive when N is large. However, by identifying and exploiting the sparsity of the Hessian, we are able to apply the conjugate-gradient method and enjoy the fast (quadratic) convergence of Newton's method at a moderate computational cost per iteration.
  - —An approach based on successive unitary transformations involving multiplications by parameterized lower and upper diagonal matrices. This method offers linear convergence at a reduced computational load per iteration.
- Empirical comparison of the proposed algorithms and their resulting performance in simulation experiments in the respective applications contexts.

The remainder of this paper is structured as follows. In the two following subsections we briefly elaborate on the two different contexts of SeDJoCo: ML source separation and CBF. In Section II we consider some theoretical issues, outlining equivalent problem-formulations and proving the existence of a solution for any set of symmetric or Hermitian positive-definite target-matrices. In Section III we outline our two solution approaches. Simulation results are presented in Section IV, demonstrating and comparing typical convergence patterns, as well as attainable sum-rates in CBF and separation performance in complex-valued source separation. The paper is concluded in Section V.

A. Motivation in ML Blind or Semiblind Source Separation

Consider the problem of blind (or semiblind) source separation, in which N statistically independent, zero-mean wide-sense stationary (and real-valued) source signals $\boldsymbol{s}[t] \triangleq [s_1[t], \ldots, s_N[t]]^T$ (with different spectra) are mixed by an unknown, square invertible (real-valued) mixing-matrix $\boldsymbol{A}$, yielding the N mixture signals $\boldsymbol{x}[t] \triangleq [x_1[t], \ldots, x_N[t]]^T$

$$\boldsymbol{x}[t] = \boldsymbol{A}\boldsymbol{s}[t], \quad t = 1, 2, \ldots, T. \tag{1}$$

When the power spectral densities (PSDs) of the sources $h_1(\nu), \ldots, h_N(\nu)$ (respectively) are known, the scenario is called "semiblind." When the PSDs are unknown, the scenario is "fully blind," see, e.g., [13]. In either case, consider some presumed PSDs (either the true PSDs in a semiblind scenario or some "educated guess" in a fully blind scenario) $\hat{h}_1(\nu), \ldots, \hat{h}_N(\nu)$, and denote by $\hat{\phi}_n[t]$ the inverse discrete-time Fourier transform (IDTFT) of $\hat{h}_n^{-1}(\nu)$, namely

$$\hat{\phi}_n[t] \triangleq \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\hat{h}_n(\nu)} \cdot e^{j2\pi\nu t} d\nu, \quad n \in \{1, 2, \ldots, N\}. \tag{2}$$

It is shown in [11] (see also [14, Ch. 7]) that for ML (in the semiblind scenario, assuming Gaussian sources) or quasi-ML (QML) (in the fully blind scenario) separation, the likelihood equations (often also called "estimating equations" in this context) for estimation of $\boldsymbol{A}$ from $\boldsymbol{x}[1], \ldots, \boldsymbol{x}[T]$ take the form

$$\sum_{\tau=1-T}^{T-1} \hat{\phi}_n[\tau] \boldsymbol{e}_m^T \hat{\boldsymbol{A}}^{-1} \hat{\boldsymbol{R}}[\tau] \hat{\boldsymbol{A}}^{-T} \boldsymbol{e}_n = 0 \quad \forall m \neq n,\ m, n \in \{1, 2, \ldots, N\} \tag{3}$$

where $\boldsymbol{e}_n$ denotes the nth column of the $N \times N$ identity matrix $\boldsymbol{I}$, and where $\boldsymbol{R}[\tau]$ denotes the observations' empirical (biased) correlation matrix estimate at lag τ,

$$\hat{\boldsymbol{R}}[\tau] \triangleq \frac{1}{T} \sum_{t=\max(1,1-\tau)}^{\min(T,T-\tau)} \boldsymbol{x}[t] \boldsymbol{x}^T[t+\tau]. \tag{4}$$

Thus, defining the set of N matrices

$$\boldsymbol{Q}_n \triangleq \sum_{\tau=1-T}^{T-1} \hat{\phi}_n[\tau] \hat{\boldsymbol{R}}[\tau], \quad n = 1, 2, \ldots, N \tag{5}$$

and denoting $\hat{\boldsymbol{B}} \triangleq \hat{\boldsymbol{A}}^{-1}$ (the ML or QML estimate of the demixing matrix), we observe that the likelihood (3) can also take the form

$$\boldsymbol{e}_m^T (\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T) \boldsymbol{e}_n = 0 \quad \forall m \neq n,\ m, n \in \{1, 2, \ldots, N\} \tag{6}$$

which implies that for each $n \in \{1, 2, \ldots, N\}$, all off-diagonal elements in the nth column of the transformed matrix $\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T$ should be zeros. It is straightforward to show that a "symmetrized" version of $\hat{\boldsymbol{R}}[\tau]$ (a result of averaging $\hat{\boldsymbol{R}}[\tau]$ with $\hat{\boldsymbol{R}}^T[\tau]$) can also be used in (3), in which case the resulting matrices $\boldsymbol{Q}_n$ would also be symmetric, and the form (6) would imply that all off-diagonal elements in both the nth column and nth row of $\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T$ must be all-zeros (for each $n \in \{1, 2, \ldots, N\}$).

It is also shown [11], [14] that an additional likelihood equation (related to the scaling of the reconstructed sources) requires that the expression in (3) equal 1 for $n = m$. Consequently, the respective diagonal (n, n)th element of $\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T$ should equal 1 as well—but this is merely a scaling condition, which may be substituted with other scaling constraints if desired. As we shall see in the sequel, this scaling constraint is used in the context of BSS, but is not applicable in other contexts, such as our CBF application. Note that all other elements (in columns and rows other than the nth) of $\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T$ are irrelevant to the ML (or QML) solution, namely, the resulting structure of each $\hat{\boldsymbol{B}} \boldsymbol{Q}_n \hat{\boldsymbol{B}}^T$ may generally be far from diagonality, as long as its nth row and column are *exactly* of the form expected in a diagonal matrix.

A similar form of estimating equations is encountered in a somewhat more specific context of Gaussian AR sources in [11] (see also [14, Ch.7]), and in a more general context (of Gaussian source signals which are not necessarily stationary, but have general temporal-covariance patterns) in [13]. The complex-valued version would also be encountered in these contexts (with complex-valued sources), but only when all signals in question are *circular* complex-valued random processes.

General AJD is basically an *ad-hoc* tool which attempts to "best fit" a prescribed model to the set of target-matrices, with no claim of optimality in any significant sense. As shown in [8], [15], in some particular cases general AJD can be made asymptotically optimal by the introduction of proper weighting. However, the same asymptotic optimality appears in a much more "natural" and computationally simpler way in SeDJoCo (with particular choices of target-matrices), since SeDJoCo can directly attain the ML estimate of $\boldsymbol{A}$ or $\boldsymbol{B}$ in such cases. In fact, following [13], it can be concluded that (asymptotically) optimal separation of independent Gaussian sources with any kind of time/frequency diversity (whether stationary, nonstationary, partly stationary, and partly nonstationary, etc.) can always be attained via the solution of a SeDJoCo problem.

B. Motivation in Coordinated Beamforming

The SeDJoCo problem is further motivated by the application of coordinated beamforming for a multiuser MIMO broadcast channel, when the system has a smaller number of transmit antennas than the aggregate number of receive antennas. In this case, many existing linear precoding techniques [e.g., zero-forcing (ZF) and block diagonalization (BD)] cannot be used due to the dimensionality constraint [16].

Previous CBF algorithms (e.g., [17], [18]) allow a smaller number of data streams than the number of receive antennas by jointly optimizing the combining vectors at the transmitter and receiver and enforcing zero multiuser interference (MUI) at each receiver. However, iterative computations are required to update the transmit beamformer and receive beamformer alternately, and the convergence of these iterative algorithms cannot be guaranteed. In order to avoid these iterative computations, in [19] a closed-form expression for CBF was proposed, but it is only valid for a system with $M_{\mathrm{T}} = 2$ transmit antennas and $K = 2$ users. We show that this task can be transformed into a SeDJoCo problem, and the coordinated transmit-receive beamformers can be calculated directly for an arbitrary number of transmit antennas M_{T}. Consequently, this SeDJoCo-based CBF can be considered as a closed-form CBF technique.

Consider a multiuser MIMO system with a single base station (BS) and K users, where the BS is equipped with M_{T} transmit antennas and user k has M_{R_k} receive antennas. The aggregate number of receive antennas is denoted by M_{R} (i.e., $M_{\mathrm{R}} = \sum_{k=1}^{K} M_{\mathrm{R}_k}$).

The number of transmit antennas is denoted M_{T}, and we shall assume $M_{\mathrm{T}} = K \leq M_{\mathrm{R}}$ (for reasons that are explained later on). The propagation channel between the BS and each user is assumed to be flat fading. The matrix $\boldsymbol{H}_k \in \mathbb{C}^{M_{\mathrm{R}_k} \times M_{\mathrm{T}}}$ represents the channel between the BS and the kth user. Let x_k denote the transmit signal for the kth user, and $\boldsymbol{b}_k \in \mathbb{C}^{M_{\mathrm{T}} \times 1}$ indicate the unit-norm transmit beamformer. Denoting the receive combining vector for user k by $\boldsymbol{w}_k \in \mathbb{C}^{M_{\mathrm{R}_k} \times 1}$, and restricting our attention to one data stream per user, the received signal of the kth user after receive combining is given by

$$y_k = \boldsymbol{w}_k^H \boldsymbol{H}_k \boldsymbol{b}_k x_k + \boldsymbol{w}_k^H \boldsymbol{H}_k \sum_{\substack{\ell=1 \\ \ell \neq k}}^{K} \boldsymbol{b}_\ell x_\ell + \boldsymbol{w}_k^H \boldsymbol{v}_k \tag{7}$$

where $\boldsymbol{v}_k \in \mathbb{C}^{M_{\mathrm{R}_k} \times 1}$ denotes the additive, zero-mean complex-valued white noise vector present at the kth receiver.

The maximum ratio combinations (matched filters) at the receivers are given by $\boldsymbol{w}_k = \boldsymbol{H}_k \boldsymbol{b}_k$. The coordinated transmission strategies choose the transmit beamforming and receive combining vectors such that each user experiences zero MUI. This implies that for the kth user $\boldsymbol{w}_k^H \boldsymbol{H}_k \boldsymbol{b}_\ell = 0$ for all $\ell \neq k$, which is equivalent to $\boldsymbol{b}_k^H \boldsymbol{H}_k^H \boldsymbol{H}_k \boldsymbol{b}_\ell = 0$ $(\forall \ell \neq k)$.

If $\boldsymbol{B}^H \in \mathbb{C}^{M_{\mathrm{T}} \times K}$ denotes the combined transmit beamformers for all users and $\boldsymbol{Q}_k \in \mathbb{C}^{M_{\mathrm{T}} \times M_{\mathrm{T}}}$ denotes a scaled version of the sample correlation matrix of user k, we have

$$\boldsymbol{B}^H = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_K] \tag{8}$$

$$\boldsymbol{Q}_k = \boldsymbol{H}_k^H \boldsymbol{H}_k \tag{9}$$

and $\boldsymbol{B}\boldsymbol{Q}_k\boldsymbol{B}^H$ can be calculated as

$$\boldsymbol{B}\boldsymbol{Q}_k\boldsymbol{B}^H = \begin{bmatrix} \boldsymbol{b}_1^H\boldsymbol{Q}_k\boldsymbol{b}_1 & \cdots & \boldsymbol{b}_1^H\boldsymbol{Q}_k\boldsymbol{b}_k & \cdots & \boldsymbol{b}_1^H\boldsymbol{Q}_k\boldsymbol{b}_K \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{b}_k^H\boldsymbol{Q}_k\boldsymbol{b}_1 & \cdots & \boldsymbol{b}_k^H\boldsymbol{Q}_k\boldsymbol{b}_k & \cdots & \boldsymbol{b}_k^H\boldsymbol{Q}_k\boldsymbol{b}_K \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{b}_K^H\boldsymbol{Q}_k\boldsymbol{b}_1 & \cdots & \boldsymbol{b}_K^H\boldsymbol{Q}_k\boldsymbol{b}_k & \cdots & \boldsymbol{b}_K^H\boldsymbol{Q}_k\boldsymbol{b}_K \end{bmatrix}. \tag{10}$$

Using the zero MUI condition $\boldsymbol{b}_k^H\boldsymbol{Q}_k\boldsymbol{b}_\ell = 0$ $(\forall \ell \neq k \in \{1, 2, \ldots, K\})$, we find that the off-diagonal elements on the kth row and the kth column of $\boldsymbol{B}\boldsymbol{Q}_k\boldsymbol{B}^H$ must be zero. This indicates that the matrix $\boldsymbol{B}$ can jointly eliminate the off-diagonal elements on the kth row and the kth column of $\boldsymbol{Q}_k$ for the entire set of matrices $\{\boldsymbol{Q}_k\}_{k=1}^K$. This property allows us to directly use the SeDJoCo solution to obtain the combined transmit beamformer $\boldsymbol{B}^H$.

Here, the number of users K is the same as the number of target-matrices, and the number M_{T} of transmit antennas corresponds to the parameter N (namely, to the dimensions of the target-matrices). Normally, N would equal the number of users K: If N is smaller than K, then the number of matrices exceeds their dimensions, and generally SeDJoCo does not have a solution in such cases (meaning that the zero MUI condition cannot be met); Conversely, if N is larger than K, then the system contains inherent redundancy, and either more users can be added, or multiple data-streams can be transmitted to some (or all) of the users: for example, if the sequence of K "target-matrices" is augmented from $\{\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_K\}$ to a sequence of $N > K$ target-matrices $\{\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_K, \boldsymbol{Q}_K, \ldots, \boldsymbol{Q}_K\}$ (such that $\boldsymbol{Q}_K$ is repeated $N-K+1$ times), then following the SeDJoCo solution the Kth user would be able to receive $N-K+1$ data-streams (transmitted with the resulting $N-K+1$ different beamformers $\boldsymbol{b}_K, \ldots, \boldsymbol{b}_N$) with zero MUI. We shall therefore concentrate in the sequel on the case $N = K$. Evidently, in this case the condition $\boldsymbol{b}_k^H\boldsymbol{Q}_k\boldsymbol{b}_\ell = 0$ $(\forall \ell \neq k \in \{1, 2, \ldots, K\})$ is equivalent to (6).

The receive beamformer of each user $(\boldsymbol{w}_n = \boldsymbol{H}_n\boldsymbol{b}_n,\ n = 1, 2, \ldots, K)$ is matched to the user's effective channel. In a system where dedicated pilots are used for the downlink, each user can estimate its own receive beamformer.

II. Equivalent Formulations and Existence of a Solution

The SeDJoCo problem formulation can take several alternative, equivalent forms, each shedding a somewhat different light on the basic aspects of this problem. The three alternative formulations presented below apply both to the real-valued and complex-valued cases.

As already mentioned, in the basic SeDJoCo formulation the number of matrices K must equal the matrices' dimensions, namely $K = N$. Thus, consider N symmetric (in the real-valued case) or Hermitian symmetric (in the complex-valued case) target-matrices $\boldsymbol{Q}_1, \ldots \boldsymbol{Q}_N$, each of dimensions $N \times N$. The SeDJoCo problem can be stated as:

P1: *Given N target-matrices $\boldsymbol{Q}_1, \ldots \boldsymbol{Q}_N$, find an $N \times N$ matrix $\boldsymbol{B} = [\boldsymbol{b}_1\ \boldsymbol{b}_2\ \cdots\ \boldsymbol{b}_N]^H$, such that*

$$\boldsymbol{b}_m^H\boldsymbol{Q}_n\boldsymbol{b}_n = \delta_{mn} \quad \forall m, n \in \{1, 2, \ldots, N\} \tag{11}$$

where δ_{mn} denotes Kronecker's delta function (*which is 1 if $m = n$ and 0 otherwise*).

Equivalently, the same problem can be stated as:

P2: *Given N target-matrices $\boldsymbol{Q}_1, \ldots \boldsymbol{Q}_N$, find an $N \times N$ matrix $\boldsymbol{B}$, such that*

$$\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H\boldsymbol{e}_n = \boldsymbol{e}_n \quad \forall n \in \{1, 2, \ldots, N\}. \tag{12}$$

In other words, each transformed matrix $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$ should be *exactly* "diagonal" in its nth column (and, since it is symmetric/Hermitian, also in its nth row), in the sense that all off-diagonal elements in these row and column must be exactly zero. All other elements may take arbitrary (nonzero) values. In addition, with the problem formulations above we also require that the diagonal (n, n)th element of $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$ be 1—but this is merely a scaling constraint on the rows of $\boldsymbol{B}$—once any matrix $\boldsymbol{B}$ satisfying the exact off-diagonal equations is found, it is straightforward to simply rescale each of its rows such that $\boldsymbol{b}_n^H\boldsymbol{Q}_n\boldsymbol{b}_n = 1$, without any effect on the "n-wise diagonality" property. As we shall see in the sequel, this scaling constraint is used in the context of BSS, but is not applicable in other contexts, such as our CBF application.

Multiplying both sides of (12) by $\boldsymbol{A} = \boldsymbol{B}^{-1}$ on the left we obtain

$$\boldsymbol{Q}_n\boldsymbol{B}^H\boldsymbol{e}_n = \boldsymbol{A}\boldsymbol{e}_n \quad \Rightarrow \quad \boldsymbol{a}_n = \boldsymbol{Q}_n\boldsymbol{b}_n \quad \forall n \in \{1, 2, \ldots, N\} \tag{13}$$

where $\boldsymbol{a}_n$ denotes the nth column of $\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_N]$. In other words, the same problem can be stated as follows:

P3: *Given N target-matrices $\boldsymbol{Q}_1, \ldots \boldsymbol{Q}_N$, find two reciprocal $N \times N$ matrices $\boldsymbol{B}$ and $\boldsymbol{A} = \boldsymbol{B}^{-1}$, such that the n-th column of $\boldsymbol{A}$ is given by $\boldsymbol{Q}_n\boldsymbol{b}_n$, with $\boldsymbol{b}_n^H$ denoting the n-th row of $\boldsymbol{B}$, $n = 1, 2, \ldots, N$.*

Assuming that all target-matrices are invertible, we may also swap the roles between $\boldsymbol{B}$ and $\boldsymbol{A}$, obtaining that the nth column $\boldsymbol{b}_n$ of $\boldsymbol{B}^H$ should be given by $\boldsymbol{Q}_n^{-1}\boldsymbol{a}_n$, where $\boldsymbol{a}_n^H$ denotes the nth row of $\boldsymbol{A}^H$, $n = 1, 2, \ldots, N$. This means that the same problem formulations *P1* and *P2* above may be cast in terms of $\boldsymbol{A}^H$ (instead of $\boldsymbol{B}$) with the inverses of the target-matrices substituting the target-matrices. This implies that the "direct" and "indirect" formulations of SeDJoCo coincide: If $\boldsymbol{B}$ is the

SeDJoCo diagonalizer of $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$, then its (conjugate) transposed inverse $\boldsymbol{A}^H$ is the SeDJoCo diagonalizer of the inverse set $\boldsymbol{Q}_1^{-1}, \ldots, \boldsymbol{Q}_N^{-1}$. It is important to note that this desirable "self-reciprocity" property, is generally *not* shared by other nonorthogonal AJD algorithms. In fact, it is easy to show that this property is satisfied in nonorthogonal AJD when (and only when) the target-matrices are *exactly* jointly diagonalizable. In general, however, the target-matrices are not exactly jointly diagonalizable: In the context of BSS they are merely estimates of exactly jointly diagonalizable matrices, and therefore would almost never be themselves exactly jointly diagonalizable; Moreover, in the context of CBF there is no reason (in general) for $\boldsymbol{Q}_n = \boldsymbol{H}_n^H \boldsymbol{H}_n$ to even be close to an exactly jointly diagonalizable form. Nevertheless, the SeDJoCo solution always enjoys the "self reciprocity," reflecting some intuitively appealing kind of "self-consistency."

As obvious, e.g., from *P1*, in the real-valued (complex-valued) case, SeDJoCo requires the solution of N^2 $((2N)^2)$ real-valued equations in N^2 $((2N)^2)$ real-valued unknowns—the real-valued parameters comprising $\boldsymbol{B}$. Since these equations are nonlinear, real-valued solutions may or may not exist in general, and may or may not be unique. We shall show, however, that if all the N target-matrices are positive-definite (PD), a solution must exist (but we do not have an explicit condition for uniqueness).

Let us consider the real-valued case first. Let $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$ denote a set of (symmetric, real-valued) PD target-matrices, and let $\lambda_n > 0$ denote the smallest eigenvalue of $\boldsymbol{Q}_n$, $n = 1, \ldots, N$. Consider the function

$$C(\boldsymbol{B}) \triangleq \log|\det \boldsymbol{B}| - \frac{1}{2}\sum_{n=1}^{N} \boldsymbol{e}_n^T \boldsymbol{B} \boldsymbol{Q}_n \boldsymbol{B}^T \boldsymbol{e}_n. \tag{14}$$

For all nonsingular $\boldsymbol{B}$, $C(\boldsymbol{B})$ is obviously a continuous and differentiable function of all elements of $\boldsymbol{B}$. In addition, $C(\boldsymbol{B})$ is bounded from above:

$$\begin{aligned} C(\boldsymbol{B}) &= \log|\det \boldsymbol{B}| - \frac{1}{2}\sum_{n=1}^{N} \boldsymbol{b}_n^T \boldsymbol{Q}_n \boldsymbol{b}_n \\ &\leq \log \prod_{n=1}^{N} \|\boldsymbol{b}_n\| - \frac{1}{2}\sum_{n=1}^{N} \lambda_n \boldsymbol{b}_n^T \boldsymbol{b}_n \\ &= \frac{1}{2}\sum_{n=1}^{N} \{\log \|\boldsymbol{b}_n\|^2 - \lambda_n \|\boldsymbol{b}_n\|^2\} \\ &\leq \frac{1}{2}\sum_{n=1}^{N} \{-\log \lambda_n - 1\} \end{aligned} \tag{15}$$

where $\|\boldsymbol{b}_n\|^2 \triangleq \boldsymbol{b}_n^T \boldsymbol{b}_n$ denotes the squared norm of $\boldsymbol{b}_n$, and where we have used the properties

1) $|\det \boldsymbol{B}| \leq \prod_{n=1}^{N} \|\boldsymbol{b}_n\|$ (Hadamard's inequality);
2) $\boldsymbol{b}_n^T \boldsymbol{Q}_n \boldsymbol{b}_n \geq \lambda_n \|\boldsymbol{b}_n\|^2$; and
3) $\log x - \lambda x \leq -\log \lambda - 1$ for all $x > 0$.

Note also that $C(\boldsymbol{B})$ tends to $-\infty$ when $\boldsymbol{B}$ approaches any singular matrix, and that, in addition, $C(\boldsymbol{B})$ has the property

$$C(\alpha \cdot \boldsymbol{B}) \xrightarrow{\alpha \to \infty} -\infty \quad \forall \boldsymbol{B}. \tag{16}$$

Consequently, $C(\boldsymbol{B})$ must attain a maximum for some nonsingular $\boldsymbol{B}$. Being a smooth function of $\boldsymbol{B}$ for all nonsingular $\boldsymbol{B}$, its derivative with respect to (w.r.t.) $\boldsymbol{B}$ at the maximum point must vanish.

Indeed, differentiating $C(\boldsymbol{B})$ w.r.t. $B_{(n,m)}$ (the (n,m)th element of $\boldsymbol{B}$) and equating zero we get (for all $m, n \in \{1, 2, \ldots, N\}$)

$$\begin{aligned} \frac{\partial C(\boldsymbol{B})}{\partial B_{(n,m)}} &= A_{(m,n)} - \frac{1}{2}\sum_{k=1}^{N} 2\boldsymbol{e}_k^T \boldsymbol{E}_{nm} \boldsymbol{Q}_k \boldsymbol{B}^T \boldsymbol{e}_k \\ &= A_{(m,n)} - \sum_{k=1}^{N} \delta_{kn} \boldsymbol{e}_m^T \boldsymbol{Q}_k \boldsymbol{B}^T \boldsymbol{e}_k \\ &= A_{(m,n)} - \boldsymbol{e}_m^T \boldsymbol{Q}_n \boldsymbol{b}_n = 0 \end{aligned} \tag{17}$$

where we have used the relation

$$\frac{\partial \log|\det \boldsymbol{B}|}{\partial B_{(n,m)}} = \frac{\partial \log \det \boldsymbol{B}}{\partial B_{(n,m)}} = A_{(m,n)} \tag{18}$$

(the first equality holds for all nonsingular real-valued $\boldsymbol{B}$), and where $\boldsymbol{E}_{nm} \triangleq \boldsymbol{e}_n \boldsymbol{e}_m^T$ denotes an all-zeros matrix with an only 1 at the (n,m)th location. By concatenating these equations for $m = 1, 2, \ldots, N$ into a vector we get $\boldsymbol{a}_n = \boldsymbol{Q}_n \boldsymbol{b}_n$, which has to be satisfied for each $n = 1, 2, \ldots, N$—as required in formulation *P3* above. This means that the solution of SeDJoCo can be expressed as the maximizer of $C(\boldsymbol{B})$, which, as mentioned above, always exists when the target-matrices are all PD.

Naturally, this derivation is closely related to the fact that SeDJoCo yields the ML (or QML) estimate of the demixing matrix in some specific BSS contexts (e.g., [11]–[13]) with some specific target-matrices. However, we obtained here a more general result, which holds for *any* set of PD target-matrices, and not only for the specific matrices used for ML or QML estimation in [11]–[13].

We now turn to consider the complex-valued case. The main formal difficulty in applying the same proof to the complex-valued case stems from the fact that $C(\boldsymbol{B})$ as defined above would be a real-valued function of a complex-valued matrix, and as such would not be differentiable w.r.t. $\boldsymbol{B}$. To mitigate this difficulty, we take the well-known approach of Brandwood [20] (or van den Bos [21]), reformulating $C(\boldsymbol{B})$ as $\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)$, such that $\boldsymbol{B}$ and $\boldsymbol{B}^*$ are considered independent variables. The "*complex-gradient*" w.r.t. $\boldsymbol{B}$ is then defined as the partial derivative of $\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)$ w.r.t. $\boldsymbol{B}$, considering $\boldsymbol{B}^*$ to be constant (and this gradient equals the complex-conjugate of the similarly defined complex-gradient w.r.t. $\boldsymbol{B}^*$). At a maximum point, the complex-gradients of $\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)$ w.r.t. both $\boldsymbol{B}$ and $\boldsymbol{B}^*$ must vanish.

Indeed, define[1]

$$\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*) \triangleq \log\det \boldsymbol{B} + \log\det \boldsymbol{B}^* - \sum_{n=1}^{N} \boldsymbol{e}_n^T \boldsymbol{B} \boldsymbol{Q}_n (\boldsymbol{B}^*)^T \boldsymbol{e}_n \tag{19}$$

and assume that the target-matrices $\boldsymbol{Q}_n$ are all Hermitian and PD, denoting the smallest eigenvalue of $\boldsymbol{Q}_n$ as $\lambda_n > 0$ $(n = 1, \ldots, N)$. Using the complex-valued version of the same arguments used above in support of (15), we have

$$\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*) \leq \sum_{n=1}^{N} \{-\log \lambda_n - 1\} \tag{20}$$

[1] Although similar to (14), this expression is not meant to be interpreted as a "complex-valued generalization" of(14).

so $\tilde{C}(\boldsymbol{B},\boldsymbol{B}^*)$ is also bounded from above, and must also be maximized by some nonsingular $\boldsymbol{B}$, such that its complex-gradient w.r.t. $\boldsymbol{B}$ (and to $\boldsymbol{B}^*$) at the maximum point must vanish.

Differentiating w.r.t. $B_{(n,m)}$ we obtain

$$\begin{aligned}\frac{\partial \tilde{C}(\boldsymbol{B},\boldsymbol{B}^*)}{\partial B_{(n,m)}} &= A_{(m,n)} - 0 - \sum_{k=1}^{N} \boldsymbol{e}_k^T \boldsymbol{E}_{nm} \boldsymbol{Q}_k (\boldsymbol{B}^*)^T \boldsymbol{e}_k \\ &= A_{(m,n)} - 0 - \sum_{k=1}^{N} \delta_{kn} \boldsymbol{e}_m^T \boldsymbol{Q}_k (\boldsymbol{B}^*)^T \boldsymbol{e}_k \\ &= A_{(m,n)} - \boldsymbol{e}_m^T \boldsymbol{Q}_n \boldsymbol{b}_n = 0 \end{aligned} \tag{21}$$

where $\boldsymbol{b}_n$ is the nth column of $\boldsymbol{B}^*$, namely $\boldsymbol{b}_n^H$ is the nth row of $\boldsymbol{B}$—as defined earlier for formulations *P1* and *P3*. Differentiation w.r.t. $\boldsymbol{B}^*$ would simply yield the complex-conjugate version of the same equation. Once again, by concatenating these equations for $m = 1, 2, \ldots, N$ into a vector we get $\boldsymbol{a}_n = \boldsymbol{Q}_n \boldsymbol{b}_n$, which corresponds to *P3* for all $n \in \{1, 2, \ldots, N\}$.

We note in passing that we have shown the existence but not the uniqueness of the solution. The number of equations (including the scaling constraints) is $K(K-1) + K = K^2$, which equals the number of free parameters in $\boldsymbol{B}$. However, since the SeDJoCo equations are nonlinear, this certainly does not imply, in general, uniqueness of the solution. Indeed, with arbitrary (positive-definite) target-matrices the SeDJoCo solution might not be unique; and yet according to our experience, in the context of BSS with sufficiently long observation length T, when the target-matrices are nearly jointly diagonalizable, the solution is unique (and approximately equals the inverse of the mixing matrix).

III. Solutions of SeDJoCo

Unlike classical AJD, the SeDJoCo problem and its solutions have rarely been addressed in the literature. To the best of our knowledge, with the exception of our recent conference-papers [9], [10], so far only two different iterative algorithms have been proposed (both in the context of ML or QML BSS): One by Pham and Garat [11], which is based on multiplicative updates of $\boldsymbol{B}$, and the other by Dégerine and Zaïdi [12], which is based on alternating oblique projections w.r.t. the columns of $\boldsymbol{B}$. Both algorithms were developed for the real-valued case only, but can also be extended to the complex-valued case. A brief summary of these two algorithms can be found in [10], as well as in [14, Ch. 7].

In this section we propose two new solution approaches: One is based on Newton's method, possibly employing a conjugate-gradient solution (e.g., [22]) of the intermediate sets of sparse linear equations [10]; and the other is based on a modification of an existing LU-based nonorthogonal AJD algorithm [9]. Both algorithms will be presented for both the real-valued and complex-valued versions of the problem.

A. Solution by Newton's Method With Conjugate Gradient (NCG)

Beginning with the real-valued version, we propose to apply Newton's method for the maximization of $C(\boldsymbol{B})$ in order to solve the nonlinear equations (17). To this end, let us define the $N^2 \times 1$ gradient vector $\boldsymbol{g}$ and the $N^2 \times N^2$ Hessian matrix $\boldsymbol{H}$ as follows (some of the basic relations used in the following derivations can be found, e.g., in [23]). First, we define the indexing function $ix(m,n) \triangleq (m-1)N + n$, which determines the location of $B_{(m,n)}$ in $\mathrm{vec}(\boldsymbol{B}^T)$ (the concatenation of the columns of $\boldsymbol{B}^T$ into an $N^2 \times 1$ vector). Then, as we have already seen in (17), the elements of the gradient vector $\boldsymbol{g}$ are given (for $m, n \in \{1, 2, \ldots, N\}$) by

$$g_{ix(m,n)} \triangleq \frac{\partial C(\boldsymbol{B})}{\partial B_{(m,n)}} = A_{(n,m)} - \boldsymbol{e}_m^T \boldsymbol{Q}_n \boldsymbol{b}_n. \tag{22}$$

Put a little differently, $\boldsymbol{g}$ can also be seen as a vectorized version $\boldsymbol{g} = \mathrm{vec}\left(\boldsymbol{G}^T\right)$ of the gradient matrix

$$\boldsymbol{G} \triangleq \boldsymbol{A}^T - [\boldsymbol{Q}_1 \boldsymbol{b}_1 \ \cdots \ \boldsymbol{Q}_N \boldsymbol{b}_N]^T. \tag{23}$$

Differentiating (17) once again w.r.t. $B_{(p,q)}$ we get the elements of the Hessian $\boldsymbol{H}$ (for all $m, n, p, q \in \{1, 2, \ldots, N\}$)

$$\begin{aligned} H_{(ix(m,n),ix(p,q))} &\triangleq \frac{\partial^2 C(\boldsymbol{B})}{\partial B_{(m,n)} \partial B_{(p,q)}} \\ &= \frac{\partial}{\partial B_{(p,q)}} \left\{ A_{(n,m)} - \boldsymbol{e}_n^T \boldsymbol{Q}_m \boldsymbol{b}_m \right\} \\ &= -\boldsymbol{e}_n^T \boldsymbol{A} \boldsymbol{E}_{pq} \boldsymbol{A} \boldsymbol{e}_m \\ &\quad - \boldsymbol{e}_n^T \boldsymbol{Q}_m \boldsymbol{e}_q \cdot \delta_{mp} \\ &= -A_{(n,p)} A_{(q,m)} \\ &\quad - Q_{m(n,q)} \cdot \delta_{mp} \end{aligned} \tag{24}$$

(where we have used the relation $\partial \boldsymbol{A} = -\boldsymbol{A} \cdot \partial \boldsymbol{B} \cdot \boldsymbol{A}$). The key observation here, is that if we differentiate at $\boldsymbol{B} = \boldsymbol{I}$, then $\boldsymbol{H}$ becomes considerably sparse, since at $\boldsymbol{B} = \boldsymbol{I}$ we also have $\boldsymbol{A} = \boldsymbol{I}$, so $A_{(n,p)} A_{(q,m)} = \delta_{np} \delta_{qm}$. The computation of the associated $N^2 \times 1$ update vector $-\boldsymbol{H}^{-1}\boldsymbol{g}$ (for updating all N^2 elements of $\boldsymbol{B}$), which apparently requires the inversion of an $N^2 \times N^2$ matrix, can then be attained with relative computational simplicity using the conjugate gradient method (which exploits this sparsity). Note, indeed, that with $\boldsymbol{B} = \boldsymbol{I}$ we have

$$\boldsymbol{H} = -\boldsymbol{P} - \mathrm{Bdiag}(\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N) \tag{25}$$

where the $\mathrm{Bdiag}(\cdot)$ operator creates a block-diagonal matrix from its matrix arguments, and where $\boldsymbol{P}$ is merely a permutation matrix transforming the $\mathrm{vec}(\cdot)$ of a matrix into the $\mathrm{vec}(\cdot)$ of its transpose, namely for any $N \times N$ matrix $\boldsymbol{Y}$, we have $\boldsymbol{P} \cdot \mathrm{vec}(\boldsymbol{Y}) = \mathrm{vec}(\boldsymbol{Y}^T)$ (note also that $\boldsymbol{P} = \boldsymbol{P}^T = \boldsymbol{P}^{-1}$).

Therefore, the operation of $\boldsymbol{H}$ on any vectorized $N \times N$ matrix $\boldsymbol{Y}^T$ can be easily expressed as

$$\boldsymbol{H} \cdot \mathrm{vec}(\boldsymbol{Y}^T) = -\mathrm{vec}\left([\boldsymbol{Q}_1 \boldsymbol{y}_1 \cdots \boldsymbol{Q}_N \boldsymbol{y}_N]^T + \boldsymbol{Y} \right) \tag{26}$$

where $\boldsymbol{y}_1, \ldots \boldsymbol{y}_N$ denote the columns of $\boldsymbol{Y}^T$ (rows of $\boldsymbol{Y}$). This relatively simple relation, requiring N^3 rather than N^4 multiplications, can be conveniently exploited in a conjugate-gradient-based computation of $\boldsymbol{H}^{-1}\mathrm{vec}(\boldsymbol{G}^T)$.

Luckily, the joint congruence structure of the SeDJoCo problem enables us to always work in the vicinity of $\boldsymbol{B} = \boldsymbol{I}$, as each update of $\boldsymbol{B}$ can be translated into a transformation of the target-matrices, defining a "new" problem in terms of the transformed matrices. In other words, suppose that a set of target-matrices $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$ is given, and that an initial guess for $\boldsymbol{B}$ is $\boldsymbol{B}^{(0)} = \boldsymbol{I}$. Following a single iteration of the Newton algorithm at $\boldsymbol{B}^{(0)} = \boldsymbol{I}$, a correction matrix $\boldsymbol{\Delta}$ is found and used for updating $\boldsymbol{B}^{(1)} = \boldsymbol{B}^{(0)} + \boldsymbol{\Delta} = \boldsymbol{I} + \boldsymbol{\Delta}$. Apparently, the next step would be to apply the next iteration of the Newton algorithm by calculating the correction matrix at $\boldsymbol{B}^{(1)}$ (so as to obtain $\boldsymbol{B}^{(2)}$), but this would no longer be computationally appealing, since at $\boldsymbol{B}^{(1)} \neq \boldsymbol{I}$ the structure of the Hessian severely departs from (26) and becomes cumbersome and nonsparse. Fortunately, an attractive alternative exists in SeDJoCo: Rather than computing the next update at $\boldsymbol{B}^{(1)}$ with the original target-matrices, transform these matrices into a new set of target-matrices, using the congruence transformation implied by $\boldsymbol{B}^{(1)}$, namely obtain $\tilde{\boldsymbol{Q}}_n = \boldsymbol{B}^{(1)}\boldsymbol{Q}_n(\boldsymbol{B}^{(1)})^T$ (for $n = 1, 2, \ldots N$). This transformation fully accounts for the update in $\boldsymbol{B}$, so that with the new set $\tilde{\boldsymbol{Q}}_1, \tilde{\boldsymbol{Q}}_2, \ldots, \tilde{\boldsymbol{Q}}_N$, once again $\boldsymbol{B}^{(0)} = \boldsymbol{I}$ would be used as an "initial guess" (consistent with the result of the previous step in the Newton algorithm), leading to a convenient calculation of the next update. The process proceeds by retransforming the new target-matrices at each step, and accumulating the updates by applying the respective left-multiplicative updates of $\boldsymbol{B}$ (such that the resulting $\boldsymbol{B}$ is the solution for the original set).

Summarizing our algorithm, given the target-matrices and some initial guess of $\boldsymbol{B}$, we repeat the following until convergence

1) Update the transformed target-matrices

$$\tilde{\boldsymbol{Q}}_n \leftarrow \boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^T \quad n = 1, 2, \ldots N.$$

2) Using (23), construct the gradient matrix $\boldsymbol{G}$ at $\boldsymbol{B} = \boldsymbol{I}$,

$$\boldsymbol{G} = \boldsymbol{I} - [\tilde{\boldsymbol{Q}}_1\boldsymbol{e}_1 \cdots \tilde{\boldsymbol{Q}}_N\boldsymbol{e}_N]^T.$$

3) Find the correction matrix $\boldsymbol{\Delta}$, given by

$$\mathrm{vec}(\boldsymbol{\Delta}^T) = -\boldsymbol{H}^{-1} \cdot \mathrm{vec}(\boldsymbol{G}^T).$$

Note: A key observation here is that the associated system of linear equations $\boldsymbol{H} \cdot \mathrm{vec}(\boldsymbol{\Delta}^T) = -\mathrm{vec}(\boldsymbol{G}^T)$ may be conveniently solved using the conjugate-gradient or the conjugate-gradient-squared method[2] (e.g., [22]). Since $\boldsymbol{H}$ is of dimensions $N^2 \times N^2$, a direct solution may be computationally too expensive for large values of N. In such cases, the sparsity of $\boldsymbol{H}$ calls for employing the conjugate-gradient method, an iterative solution with guaranteed convergence in a finite number of steps. The method does not involve explicit inversion of $\boldsymbol{H}$, but merely requires computation of products of the form $\boldsymbol{H} \cdot \boldsymbol{y}$ in each iteration. As shown in (26) above, such products can be computed with N^3 (rather than N^4) multiplications, by exploiting the special sparse structure of $\boldsymbol{H}$.

4) Apply and accumulate the correction $\boldsymbol{B} \leftarrow (\boldsymbol{I} + \boldsymbol{\Delta})\boldsymbol{B}$

This algorithm is somewhat similar in structure to Pham's multiplicative updates algorithm [11]. However, it is based on an iterative solution of (17), which, unlike the direct solution of (11) (used in [11]), conveniently lends itself to the use of a conjugate gradient algorithm in each Newton iteration, by exploiting the sparsity of $\boldsymbol{H}$. We note further that the multiplicative updates algorithm in [11] assumes, for further simplification, that at the vicinity of a solution the transformed matrices $\hat{\boldsymbol{B}}\boldsymbol{Q}_n\hat{\boldsymbol{B}}^T$ are all nearly diagonal. While this assumption may be reasonable in the context of BSS (since near separation the empirical correlation matrices are all nearly diagonal if the observation length T is sufficiently long), it excludes non-BSS applications (such as our proposed CBF), in which there is no reason for the transformed matrices to exhibit any diagonality on top of the attained "n-wise diagonality."

In order to extend the algorithm to the complex-valued case, we must recall once again that for complex-valued $\boldsymbol{B}$, the gradient and the Hessian of the real-valued $C(\boldsymbol{B})$ w.r.t. $\boldsymbol{B}$ are undefined. We must therefore resort once again to van den Bos' "complex-gradient" and "complex-Hessian" [21], and apply Newton's approach to the maximization of (19). To this end, we need:

- The gradient of $\tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)$ w.r.t. $\boldsymbol{B}$, which we shall denote in vector form as the $N \times 1$ vector $\boldsymbol{g}^{\circ}$;
- The gradient w.r.t. $\boldsymbol{B}^*$, which we shall denote $\boldsymbol{g}^*$;
- The Hessian w.r.t. $\boldsymbol{B}$ and $\boldsymbol{B}$, which we shall denote by the $N^2 \times N^2$ matrix $\boldsymbol{H}^{\circ\circ}$;
- The Hessian w.r.t. $\boldsymbol{B}$ and $\boldsymbol{B}^*$, which we shall denote $\boldsymbol{H}^{\circ *}$;
- The Hessian w.r.t. $\boldsymbol{B}^*$ and $\boldsymbol{B}$, which we shall denote $\boldsymbol{H}^{*\circ}$;
- The Hessian w.r.t. $\boldsymbol{B}^*$ and $\boldsymbol{B}^*$, which we shall denote $\boldsymbol{H}^{**}$;

Evidently,

$$g^{\circ}_{ix(m,n)} \triangleq \frac{\partial \tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)}{\partial B_{(m,n)}} = A_{(n,m)} - \boldsymbol{e}_n^T\boldsymbol{Q}_m\boldsymbol{b}_m \tag{27}$$

which is a vectorized version $\boldsymbol{g}^{\circ} = \mathrm{vec}((\boldsymbol{G}^{\circ})^T)$ of

$$\boldsymbol{G}^{\circ} \triangleq \boldsymbol{A}^T - [\boldsymbol{Q}_1\boldsymbol{b}_1 \cdots \boldsymbol{Q}_N\boldsymbol{b}_N]^T. \tag{28}$$

Differentiating once again w.r.t. $B_{(p,q)}$

$$\begin{aligned} H^{\circ\circ}_{(ix(m,n),ix(p,q))} &\triangleq \frac{\partial^2 \tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)}{\partial B_{(m,n)}\partial B_{(p,q)}} \\ &= \frac{\partial}{\partial B_{(p,q)}}\left\{A_{(n,m)} - \boldsymbol{e}_n^T\boldsymbol{Q}_m(\boldsymbol{B}^*)^T\boldsymbol{e}_m\right\} \\ &= -\,\boldsymbol{e}_n^T\boldsymbol{A}\boldsymbol{E}_{pq}\boldsymbol{A}\boldsymbol{e}_m - 0 \\ &= -\,A_{(n,p)}A_{(q,m)}. \end{aligned} \tag{29}$$

Conversely, differentiating w.r.t. $B^*_{(p,q)}$

$$\begin{aligned} H^{\circ *}_{(ix(m,n),ix(p,q))} &\triangleq \frac{\partial^2 \tilde{C}(\boldsymbol{B}, \boldsymbol{B}^*)}{\partial B_{(m,n)}\partial B^*_{(p,q)}} \\ &= \frac{\partial}{\partial B^*_{(p,q)}}\left\{A_{(n,m)} - \boldsymbol{e}_n^T\boldsymbol{Q}_m(\boldsymbol{B}^*)^T\boldsymbol{e}_m\right\} \\ &= 0 - \boldsymbol{e}_n^T\boldsymbol{Q}_m\boldsymbol{E}_{pq}^T\boldsymbol{e}_m \\ &= -\,\boldsymbol{e}_n^T\boldsymbol{Q}_m\boldsymbol{e}_q\delta_{mp}. \end{aligned} \tag{30}$$

Naturally, we also have $\boldsymbol{g}^* = (\boldsymbol{g}^{\circ})^*$, $\boldsymbol{H}^{**} = (\boldsymbol{H}^{\circ\circ})^*$ and $\boldsymbol{H}^{*\circ} = (\boldsymbol{H}^{\circ *})^*$.

[2] Conjugate-gradient requires $-\boldsymbol{H}$ to be PD. Although $\mathrm{Bdiag}(\tilde{\boldsymbol{Q}}_1, \ldots \tilde{\boldsymbol{Q}}_N)$ is PD, $-\boldsymbol{H}$ is generally not PD, since $\boldsymbol{P}$ is not PD. In such cases the slightly modified conjugate-gradient-squared algorithm may be used.

The vectorized update matrix is then given by

$$\begin{bmatrix}\boldsymbol{\delta}\\ \boldsymbol{\delta}^*\end{bmatrix} = -\begin{bmatrix}\boldsymbol{H}^{\circ\circ} & \boldsymbol{H}^{\circ *}\\ \boldsymbol{H}^{*\circ} & \boldsymbol{H}^{**}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{g}^{\circ}\\ \boldsymbol{g}^*\end{bmatrix}. \tag{31}$$

Obviously, it is sufficient to solve for the first half only. Using the four-blocks matrix inversion relation, the solution for $\boldsymbol{\delta}$ is also given by

$$\boldsymbol{\delta} = -[\boldsymbol{H}^{\circ\circ} - \boldsymbol{H}^{\circ *}(\boldsymbol{H}^{\circ\circ})^{-1}\boldsymbol{H}^{*\circ}]^{-1}[\boldsymbol{g}^{\circ} - \boldsymbol{H}^{\circ *}(\boldsymbol{H}^{\circ\circ})^{-1}\boldsymbol{g}^*]. \tag{32}$$

In order to simplify, we once again take advantage of the ability to work at $\boldsymbol{B} = \boldsymbol{I}$. Substituting $\boldsymbol{B} = \boldsymbol{I}$ (and $\boldsymbol{A} = \boldsymbol{I}$) for the Hessian matrices (29),(30), we get

$$\begin{aligned}\boldsymbol{H}^{\circ\circ} &= \boldsymbol{H}^{**} = -\boldsymbol{P}\ ,\\ \boldsymbol{H}^{\circ *} &= (\boldsymbol{H}^{*\circ})^* = -\mathrm{Bdiag}(\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_N),\end{aligned} \tag{33}$$

so that (32) reduces into

$$\boldsymbol{\delta} = [\boldsymbol{P} - \boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{\Lambda}^*]^{-1}[\boldsymbol{g}^{\circ} - \boldsymbol{\Lambda}^*\boldsymbol{P}^T\boldsymbol{g}^*]. \tag{34}$$

where $\boldsymbol{\Lambda} \triangleq \mathrm{Bdiag}\{\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_N\}$ is used as a shorthand notation.

The conjugate gradient method can be used here as well, by exploiting the sparsity of the complete Hessian matrix (at $\boldsymbol{B} = \boldsymbol{I}$), since for any $N \times N$ (complex-valued) matrix $\boldsymbol{Y}$, the product

$$\begin{aligned}\begin{bmatrix}-\boldsymbol{P} & -\boldsymbol{\Lambda}\\ -\boldsymbol{\Lambda}^* & -\boldsymbol{P}\end{bmatrix}&\begin{bmatrix}\mathrm{vec}(\boldsymbol{Y}^T)\\ \mathrm{vec}(\boldsymbol{Y}^H)\end{bmatrix}\\ &= -\begin{bmatrix}\mathrm{vec}\left(\boldsymbol{Y} + [\boldsymbol{Q}_1\boldsymbol{y}_1 \cdots \boldsymbol{Q}_N\boldsymbol{y}_N]^T\right)\\ \mathrm{vec}\left(\boldsymbol{Y}^* + [\boldsymbol{Q}_1\boldsymbol{y}_1 \cdots \boldsymbol{Q}_N\boldsymbol{y}_N]^H\right)\end{bmatrix}\end{aligned} \tag{35}$$

can be computed in N^3 instead of $(2N)^4$ (complex-valued) multiplications.

To summarize, the complex-valued version of the NCG algorithm takes the following form: Given the target-matrices $\boldsymbol{Q}_n$, $n = 1, 2, \dots, N$ and an initial guess of $\boldsymbol{B}$, repeat the following until convergence

1) Update the transformed target-matrices

$$\tilde{\boldsymbol{Q}}_n \leftarrow \boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H \quad n = 1, 2, \dots N.$$

2) Using (28), construct the gradient matrix $\boldsymbol{G}^{\circ}$ at $\boldsymbol{B} = \boldsymbol{I}$

$$\boldsymbol{G}^{\circ} = \boldsymbol{I} - [\tilde{\boldsymbol{Q}}_1\boldsymbol{e}_1 \cdots \tilde{\boldsymbol{Q}}_N\boldsymbol{e}_N]^T,$$

and denote $\boldsymbol{g}^{\circ} = \mathrm{vec}((\boldsymbol{G}^{\circ})^T)$.

3) Find the correction matrix $\boldsymbol{\Delta}$, given by

$$\mathrm{vec}(\boldsymbol{\Delta}^T) = [\boldsymbol{P} - \boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{\Lambda}^*]^{-1}[\boldsymbol{g}^{\circ} - \boldsymbol{\Lambda}^*\boldsymbol{P}^T(\boldsymbol{g}^{\circ})^*]$$

with $\boldsymbol{\Lambda} = \mathrm{Bdiag}\left(\tilde{\boldsymbol{Q}}_1, \dots, \tilde{\boldsymbol{Q}}_N\right)$. To alleviate the computational load, the conjugate-gradient method can be used for this part, exploiting the sparsity of the complex Hessian by the use of (35) (with each $\boldsymbol{Q}_n$ substituted by $\tilde{\boldsymbol{Q}}_n$).

4) Apply and accumulate the correction $\boldsymbol{B} \leftarrow (\boldsymbol{I} + \boldsymbol{\Delta})\boldsymbol{B}$

We emphasize in passing that although the scaling equations $\boldsymbol{b}_n^T\boldsymbol{Q}_n\boldsymbol{b}_n = 1$ (or $\boldsymbol{b}_n^H\boldsymbol{Q}_n\boldsymbol{b}_n = 1$) are inherently built into the NCG algorithm above, they are sometimes irrelevant. For example, in the CBF application this scaling constraint does not apply, since a unit-power constraint in the transmission translates into $\boldsymbol{b}_n^H\boldsymbol{b}_n = 1$, and a large value of $\boldsymbol{b}_n^T\boldsymbol{Q}_n\boldsymbol{b}_n$ leads to a large power of the desired $n\mathrm{th}$ signal at the receiver. Clearly, if a matrix $\boldsymbol{B}$ solves the SeDJoCo problem with any scaling equations, then for any diagonal matrix $\boldsymbol{D}$, $\boldsymbol{D} \cdot \boldsymbol{B}$ also solves SeDJoCo, but with possibly different scaling equations. Therefore, the NCG solution can be used with different scaling constraints, simply by renormalizing the rows of $\boldsymbol{B}$ as desired.

B. Solution by Structured Joint Congruence (STJOCO) Transformation

Another approach for solving SeDJoCo is to employ a modified version of an existing AJD algorithm. To this end, we propose a method based on modifying Afsari's LU-based nonorthogonal matrix joint diagonalization [24]. Our proposed approach is given the acronym structured joint congruence (STJOCO) transformation [9].

Our goal in the successive minimization approach is to find a matrix $\boldsymbol{B}$ which minimizes the magnitudes of the off-diagonal elements in the nth row and the nth column of the nth transformed target-matrix, $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$. Unlike NCG, we shall ignore the scaling constraint, using the following criterion for minimization, which is based on a modification of Afsari's scaling-invariant cost-function [24]:

$$J(\boldsymbol{B}) = \sum_{n=1}^{N}\left\|\boldsymbol{Q}_n - \boldsymbol{B}^{-1}\left[\mathrm{Diag}(\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H) + \boldsymbol{G}_n\right]\boldsymbol{B}^{-H}\right\|_{\mathrm{F}}^2 \tag{36}$$

where $\boldsymbol{G}_n = \boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$, except for its diagonal elements and the off-diagonal elements of its nth row and nth column, which are all set to zeros , and where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. We introduce an LU-based algorithm using triangular Jacobi matrices for the minimization of J. The matrix $\boldsymbol{B}$ is updated iteratively in the following manner:

$$\boldsymbol{B} \leftarrow (\boldsymbol{I} + \boldsymbol{\Delta})\boldsymbol{B} = \boldsymbol{L} \cdot \boldsymbol{U} \cdot \boldsymbol{B} \tag{37}$$

where $\mathrm{diag}(\boldsymbol{\Delta}) = \boldsymbol{0}$, and $\boldsymbol{\Delta}$ is sought so as to minimize the resulting $J(\boldsymbol{B})$. However, instead of finding $(\boldsymbol{I} + \boldsymbol{\Delta})$ explicitly, STJOCO employs an iterative LU-based algorithm, and finds $\boldsymbol{L}$ and $\boldsymbol{U}$ separately, so as to maximally reduce the resulting $J(\boldsymbol{B})$ at each step. The matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ are $N \times N$ unit lower and upper triangular matrices, respectively. Here a unit triangular matrix is a triangular matrix with diagonal elements of one. Unit lower and upper triangular matrices of dimension $N \times N$ form Lie groups denoted by $\mathcal{L}(N)$ and $\mathcal{U}(N)$, respectively. This fact simplifies the minimization process significantly, since any element of $\mathcal{L}(N)$ or $\mathcal{U}(N)$ can be represented as a product of unit lower or upper triangular Jacobi matrices, namely of lower or upper triangular matrices with only one nonzero off-diagonal element in each. Then, the $\frac{N(N-1)}{2}$-dimensional minimization problem of finding $\boldsymbol{L}$ or $\boldsymbol{U}$ so as to minimize J can be decomposed into a sequence of one-dimensional problems of finding a unit lower or upper triangular Jacobi matrix for minimizing J.

1) The Case of Real-Valued Symmetric Matrices: Let us define $\boldsymbol{L}_{m,n}(a)$ as a unit lower triangular Jacobi matrix with parameter $a \in \mathbb{R}$ corresponding to the position $(m, n), m > n$

and the rest of its off-diagonal entries are zeros. $\boldsymbol{L}_{m,n}(a)$ is an element of $\mathcal{L}(N)$. Similarly, we define a unit upper triangular Jacobi matrix with parameter a corresponding to the position $(m,n), m < n$ as $\boldsymbol{U}_{m,n}(a)$, which is an element of $\mathcal{U}(N)$. Now the minimization problem is a sequence of one-dimensional problems of finding the parameter a of a triangular Jacobi matrix $\boldsymbol{L}_{m,n}(a)$ or $\boldsymbol{U}_{m,n}(a)$ for minimizing J. We propose a simple lemma to solve the one-dimensional problem.

Lemma 1: For both $\boldsymbol{L}_{m,n}(a)$ with $m > n$, and $\boldsymbol{U}_{m,n}(a)$ with $m < n$, the respective cost-functions $J(\boldsymbol{L}_{m,n}(a))$ and $J(\boldsymbol{U}_{m,n}(a))$ can be expressed as fourth-order polynomials in a

$$\begin{aligned} J(\boldsymbol{U}_{m,n}(a)) &= J(\boldsymbol{L}_{m,n}(a)) \\ &= b_4 a^4 + b_3 a^3 + b_2 a^2 + b_1 a + b_0 \end{aligned} \tag{38}$$

with coefficients

$$\begin{aligned} b_4 &= 4 \sum_{k \in \{m,n\}} Q_k^2(n,n) \\ b_3 &= 8 \sum_{k \in \{m,n\}} Q_k(n,n) Q_k(m,n) \\ b_2 &= \sum_{k \in \{m,n\}} \sum_{\substack{p=1 \\ p \neq m,n}}^{N} \Big[2\, Q_k^2(n,n) + 4\, Q_k^2(m,n) \\ &\qquad + 2\, Q_k^2(n,p) \Big] \\ b_1 &= 4 \sum_{k \in \{m,n\}} Q_k(m,n) Q_k(n,n) \\ &\quad + 4 \sum_{\substack{p=1 \\ p \neq m,n}}^{N} Q_m(m,p) Q_m(n,p) \\ b_0 &= 2 \sum_{k=1}^{N} \sum_{\substack{p=1 \\ p \neq k}}^{N} Q_k^2(k,p) . \end{aligned}$$

Here $Q_k(\mathrm{index}_1, \mathrm{index}_2)$ denotes the $(\mathrm{index}_1, \mathrm{index}_2)$ element of the matrix $\boldsymbol{Q}_k$.

Notice that by definition $J(\boldsymbol{L}_{m,n}(a))$ and $J(\boldsymbol{U}_{m,n}(a))$ are always nonnegative. For a small a (i.e., $|a| < 1$) J is convex on $\mathbb{R}$ and we can always find a global minimum by solving the cubic polynomial $\frac{\partial J(\boldsymbol{L}_{m,n}(a))}{\partial a} = 0$ or $\frac{\partial J(\boldsymbol{U}_{m,n}(a))}{\partial a} = 0$. As a result, the value of the cost -function J is reduced at each step. Note that for the minimization of $J(\boldsymbol{L}_{m,n}(a))$ and $J(\boldsymbol{U}_{m,n}(a))$, a only depends on the elements of the matrices $\boldsymbol{Q}_m$ and $\boldsymbol{Q}_n$. The STJOCO procedure is summarized here.

1) Set $\boldsymbol{B} = \boldsymbol{I}$ and set a threshold ϵ.
2) For $m = 1, \ldots, N$ and $n = 1, \ldots, N$ $(n \neq m)$
   - Upper triangular part $(m < n)$: set $\boldsymbol{U} = \boldsymbol{I}$
     — find a such that $J(\boldsymbol{U}_{m,n}(a))$ is minimized according to Lemma 1.
     — Update all $\boldsymbol{Q}_k$ $(k \in \{1, 2, \ldots, K\})$ and $\boldsymbol{U}$ by setting
     $$\begin{aligned} \boldsymbol{Q}_k &\leftarrow \boldsymbol{U}_{m,n}(a) \boldsymbol{Q}_k \boldsymbol{U}_{m,n}(a)^T \\ \boldsymbol{U} &\leftarrow \boldsymbol{U}_{m,n}(a) \boldsymbol{U} . \end{aligned}$$
   - Lower triangular part $(m > n)$: set $\boldsymbol{L} = \boldsymbol{I}$
     — find a such that $J(\boldsymbol{L}_{m,n}(a))$ is minimized according to Lemma 1.
     — Update all $\boldsymbol{Q}_k$ $(k \in \{1, 2, \ldots, K\})$ and $\boldsymbol{L}$ by setting
     $$\begin{aligned} \boldsymbol{Q}_k &\leftarrow \boldsymbol{L}_{m,n}(a) \boldsymbol{Q}_k \boldsymbol{L}_{m,n}(a)^T \\ \boldsymbol{L} &\leftarrow \boldsymbol{L}_{m,n}(a) \boldsymbol{L}. \end{aligned}$$
3) Update $\boldsymbol{B}$ by setting $\boldsymbol{B} \leftarrow \boldsymbol{L}\boldsymbol{U}\boldsymbol{B}$. If $\frac{J(\boldsymbol{B}_q) - J(\boldsymbol{B}_{q+1})}{J(\boldsymbol{B}_q)} > \epsilon$, then go to step 2. Otherwise, the procedure has ended.

We can also use other stopping criteria such as tracking the changes in $\boldsymbol{B}$ (e.g., $\|\boldsymbol{L}\boldsymbol{U} - \boldsymbol{I}\|_{\mathrm{F}}$).

2) The Case of Complex-Valued Hermitian Matrices: We define $\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi))$ as a unit lower triangular Jacobi matrix with $a \cdot \exp(j\varphi)$ at the position (m,n) for $m > n$, the remaining off-diagonal entries of $\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi))$ are zero. The parameters a and φ are real-valued and $a > 0$. In a similar fashion we define a unit upper triangular Jacobi matrix $\boldsymbol{U}_{m,n}(a \cdot \exp(j\varphi))$ for $m < n$. Then, we use a sequence of one dimensional minimization problems to replace the $\frac{N(N-1)}{2}$ dimensional minimization problem. However, in contrast to the real-valued case, two parameters a and φ have to be determined. We propose Lemma 2 to solve the complex one-dimensional problem.

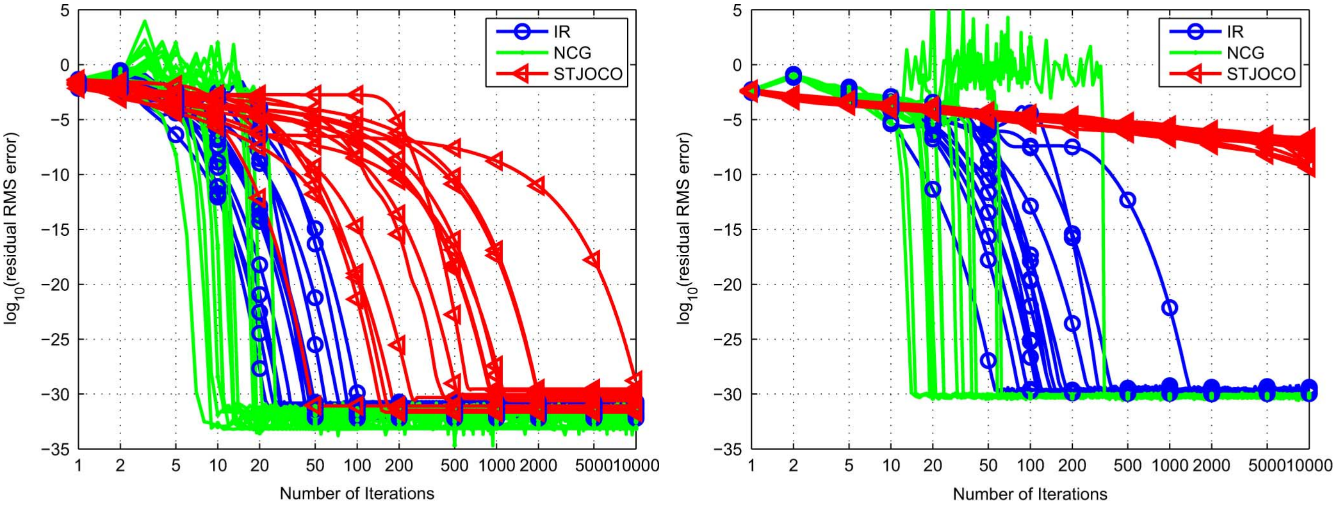
Lemma 2: For both $\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi))$ with $m > n$, and $\boldsymbol{U}_{m,n}(a \cdot \exp(j\varphi))$ with $m < n$, the respective cost-functions can be expressed as fourth-order polynomials in a (with coefficients depending on φ) as follows:

$$\begin{aligned} J(\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi))) &= c_4 a^4 + c_3(\varphi) a^3 + c_2(\varphi) a^2 \\ &\quad + c_1(\varphi) a + c_0 \\ J(\boldsymbol{U}_{m,n}(a \cdot \exp(j\varphi))) &= c_4 a^4 + c_3(-\varphi) a^3 + c_2(\varphi) a^2 \\ &\quad + c_1(-\varphi) a + c_0 \end{aligned} \tag{39}$$

with coefficients

$$\begin{aligned} c_4 &= 4 \sum_{k \in \{m,n\}} Q_k^2(n,n) \\ c_3(\varphi) &= 8 \sum_{k \in \{m,n\}} Q_k(n,n) \cdot \mathrm{Re}\left\{ Q_k(m,n) \exp(j\varphi) \right\} \\ c_2(\varphi) &= 2 \sum_{k \in \{m,n\}} \sum_{\substack{p=1 \\ p \neq m,n}}^{N} \Big[Q_k^2(n,n) + 4\, |Q_k(m,n)|^2 \\ &\qquad + |Q_k(n,p)|^2 + \mathrm{Re}\left\{ Q_k^2(n,m) \exp(j2\varphi) \right\} \Big] \\ c_1(\varphi) &= 4 \sum_{k \in \{m,n\}} Q_k(n,n) \cdot \mathrm{Re}\left\{ Q_k(n,m) \exp(j\varphi) \right\} \\ &\quad + 4 \sum_{\substack{p=1 \\ p \neq m,n}}^{N} \mathrm{Re}\left\{ Q_m(p,m) Q_m(n,p) \exp(j\varphi) \right\} \\ c_0 &= 2 \sum_{k=1}^{N} \sum_{\substack{p=1 \\ p \neq k}}^{N} |Q_k(k,p)|^2 . \end{aligned}$$

The cost-functions $J(\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi)))$ and $J(\boldsymbol{U}_{m,n}(a \cdot \exp(j\varphi)))$ are fourth-order polynomials in a and second-order polynomials in $\cos(\varphi)$. We can compute the algebraic solutions



(a)

(b)

Fig. 2. RMS error for IR, NCG, and STJOCO with arbitrary positive-definite real-valued matrices (a) for $K = N = 3$; (b) for $K = N = 10$.

for $\frac{\partial J(\boldsymbol{L}_{m,n}(a\cdot\exp(j\varphi)))}{\partial a} = 0$ and $\frac{\partial J(\boldsymbol{L}_{m,n}(a\cdot\exp(j\varphi)))}{\partial \varphi} = 0$. However, these expressions are quite complicated. Alternatively, we can employ numerical nonlinear convex optimization methods (e.g., the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method [25] with cubic line search) to find the optimal point (a, φ) which minimizes the cost-function $J(\boldsymbol{L}_{m,n}(a \cdot \exp(j\varphi)))$, since J is convex for small a and φ (i.e., $0 < a < 1$ and $|\varphi| < \frac{\pi}{4}$). In our case these two methods reach the same optimal point for (a, φ). Lemma 2 shows us that, as in the real-valued case, the minimizations of $J(\boldsymbol{L}_{m,n}(a\cdot\exp(j\varphi)))$ and $J(\boldsymbol{U}_{m,n}(a\cdot\exp(j\varphi)))$ only depends on the elements of the matrices $\boldsymbol{Q}_m$ and $\boldsymbol{Q}_n$.

The STJOCO procedure for the complex-valued case is the same as for the real-valued case, except for step 2. In step 2 we find both a and φ for the minimization of $J(\boldsymbol{L}_{m,n}(a\cdot\exp(j\varphi)))$ or $J(\boldsymbol{U}_{m,n}(a \cdot \exp(j\varphi)))$ according to Lemma 2.

IV. Simulation Results

In this section we present the results of three different simulation experiments. In the first subsection we demonstrate typical convergence patterns of our two proposed iterative algorithms (comparing also to an existing iterative algorithm) in solving the generic SeDJoCo problem with random target-matrices. We then turn to demonstrate the resulting performance in the context of the two prominent applications mentioned above: CBF in a MIMO channel in Section IV-B, and BSS of complex-valued sources in Section IV-C.

A. Convergence Behavior

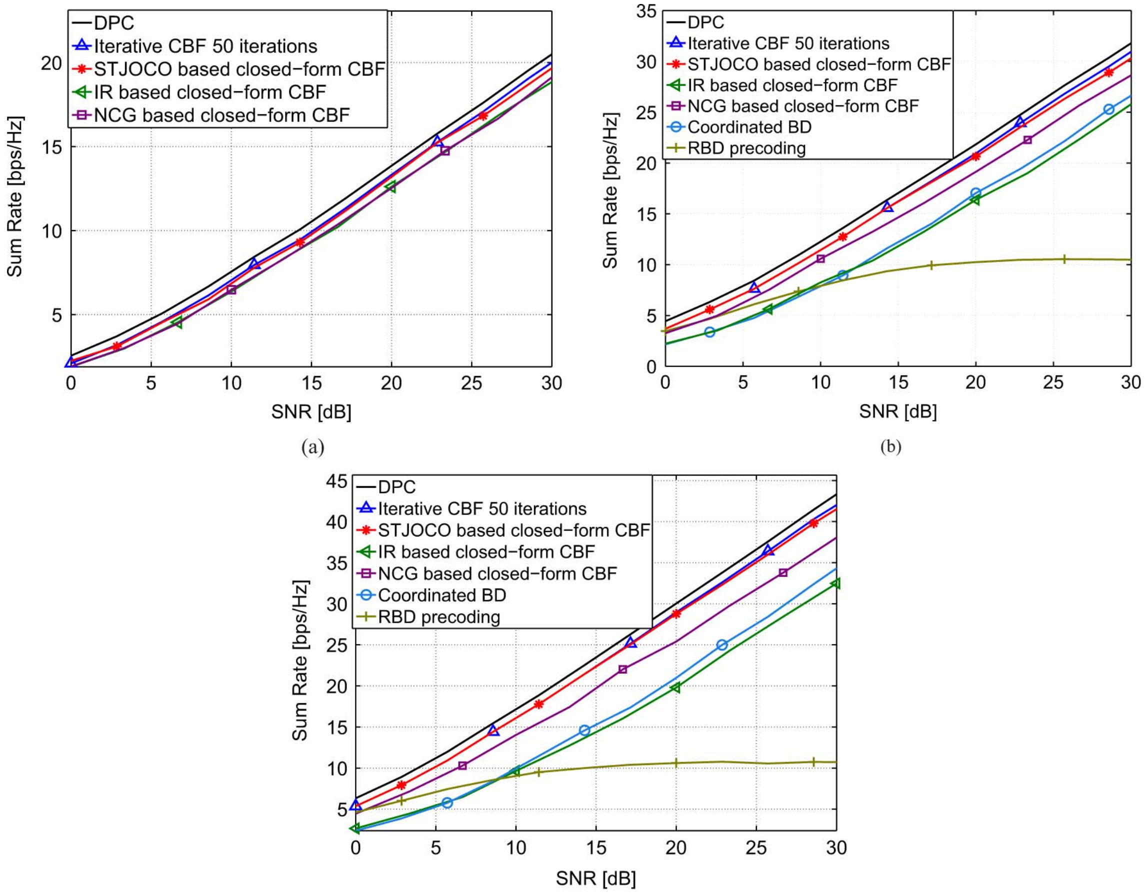
We evaluate the convergence of the proposed NCG and STJOCO solutions of the SeDJoCo problem in the terms of the logarithm of the residual root-mean-squares (RMS) error versus the iteration number. The target-matrices are arbitrary, randomly generated, symmetric, positive-definite, real-valued matrices (we obtain very similar results, not presented in here, for complex-valued matrices). The convergence is compared to the iterative relaxation (IR) algorithm proposed in [12], but not to the solution proposed in [11], since the latter does not converge, in general, for arbitrary target-matrices (but only for "nearly jointly diagonalizable" matrices). For all three solutions we initialize the sought matrix $\boldsymbol{B}$ to the identity matrix (i.e., $\boldsymbol{B} = \boldsymbol{I}$), except for the NCG algorithm with large values of N, because as N grows, the NCG algorithm becomes more sensitive to the initialization. Therefore, for NCG with $N = 10$ we initialize $\boldsymbol{B}$ to the output of the IR algorithm obtained as soon as the RMS error falls below 10^{-5} (and, for fair comparison, we continue the iteration count from the respective IR iteration number).

Fig. 2(a) and (b) shows typical convergence patterns of the three iterative algorithms (IR, NCG, and STJOCO) for several independent trials. (Note that the numbers on the *y* axis are $\log_{10}$ of the RMS error, and are *not* given in dB—the lower "saturation line" reflects an average residual error of about 10^{-30}, which means that a convergence pattern reaching that line attains the *exact* solution, which zeros-out the respective matrix-elements up to the machine-accuracy.)

It is evident that the NCG algorithm significantly accelerates the convergence: For $N = 3$, in 100 independent trials the median number of iterations until convergence to an exact solution[3] was 42 for IR, 13 for NCG, and 635 for STJOCO. For $N = 10$, that number was 127 for IR and 23 for NCG. As evident from Fig. 2(b), STJOCO did not converge to a machine-accuracy solution for $N = 10$ (with the maximal tested number of 10 000 iterations), but still attained very reasonable solutions, with small residual errors (way below 10^{-5} after more than 500 iterations)—which are probably local minima of its respective cost-function.

The accelerated convergence of NCG is obtained at the cost of only a moderate increase in the computational complexity per iteration. The complexity per iteration is $\mathcal{O}(N^4)$ for the IR algorithm and approximately $\mathcal{O}(N^5)$ for the NCG algorithm. The STJOCO algorithm has a comparable computational complexity $\mathcal{O}(N^4)$ per iteration, but occasionally converges to local nonzero minima of the cost-function $J(\boldsymbol{B})$, which are not exact

[3] We define "convergence to an exact solution" as the state where the residual RMS error drops to the machine-accuracy, around 10^{-30}.



(a)

(b)

(c)

Fig. 3. Achievable sum rate comparison (a) for $M_{\mathrm{T}} = M_{\mathrm{R}_i} = K = 2$; (b) for $M_{\mathrm{T}} = M_{\mathrm{R}_i} = K = 3$; (c) for $M_{\mathrm{T}} = M_{\mathrm{R}_i} = K = 4$.

solutions of SeDJoCo. As we shall see immediately, in the context of CBF this apparent disadvantage of STJOCO is generally compensated for by larger values of the diagonal terms in the “drilled” matrices, which implies a higher effective SINR, partly compensating for the residual MUI.

B. Achievable Sum Rate of Multiuser MIMO Application

As mentioned above, one motivation for considering the SeDJoCo problem is the application for “closed-form” coordinated beamforming in multiuser MIMO broadcast channels. In this subsection, we evaluate the performance of STJOCO and NCG in terms of the achievable sum rate of a multiuser MIMO system. Like in the previous experiment, we also compare our results to the IR solution [12]. In addition, we compare them to regularized block diagonalization (RBD) linear precoding [26], since RBD can still be applied under the condition that the system has a smaller number of transmit antennas M_{T} than the total number of receive antennas M_{R}. In the simulations, we have $M_{\mathrm{T}} = K$ and transmit one data stream to each user. For simplicity, an equal power allocation is employed among the users. The achievable sum rate is calculated as $R = \sum_{i=1}^{K} \log_2(1 + \mathrm{SINR}_i)$, where SINR_i indicates the signal-to-interference-plus-noise ratio at the user i. Dirty paper coding (DPC) has been shown to achieve the capacity region of Gaussian MIMO broadcast channels in [27]. Therefore, we use the achievable sum rate of DPC as a benchmark.

Fig. 3(a) shows the comparisons of the iterative coordinated beamforming (CBF) algorithms [17], the proposed STJOCO- and NCG-based, as well as IR-based closed-form CBF algorithms, when the system has two transmit antennas with two users and each user is equipped with two receive antennas. It is observed that the STJOCO based closed-form CBF almost achieves the same sum rate performance as the iterative CBF and performs better than NCG-based and IR-based closed-form CBF. After rescaling each column of the combined transmit beamforming matrix $\boldsymbol{B}^H$ to have unit norm, the STJOCO solution tends to yield larger diagonal elements on the “drilled” rows and columns of the transformed target-matrices (compared to the IR and NCG solutions). The larger magnitude of these diagonal elements results in an enhanced SINR, thereby leading to higher achievable sum rates, even in cases where STJOCO does not attain an exact solution and some residual MUI is present. This has been consistently observed in our simulations.

In Fig. 3(b) and (c), the comparisons of the iterative CBF algorithm [17], the proposed STJOCO- and NCG- and IR-based closed-form CBF algorithms, the suboptimal coordinated BD algorithm [16], and RBD precoding [26] are presented. We can see that the STJOCO-based closed-form CBF performs much better than the NCG- and the IR-based closed-form CBF algorithms as well as the suboptimal coordinated BD, by achieving almost the same performance as the iterative CBF. The performance of RBD is heavily degraded when the system has a

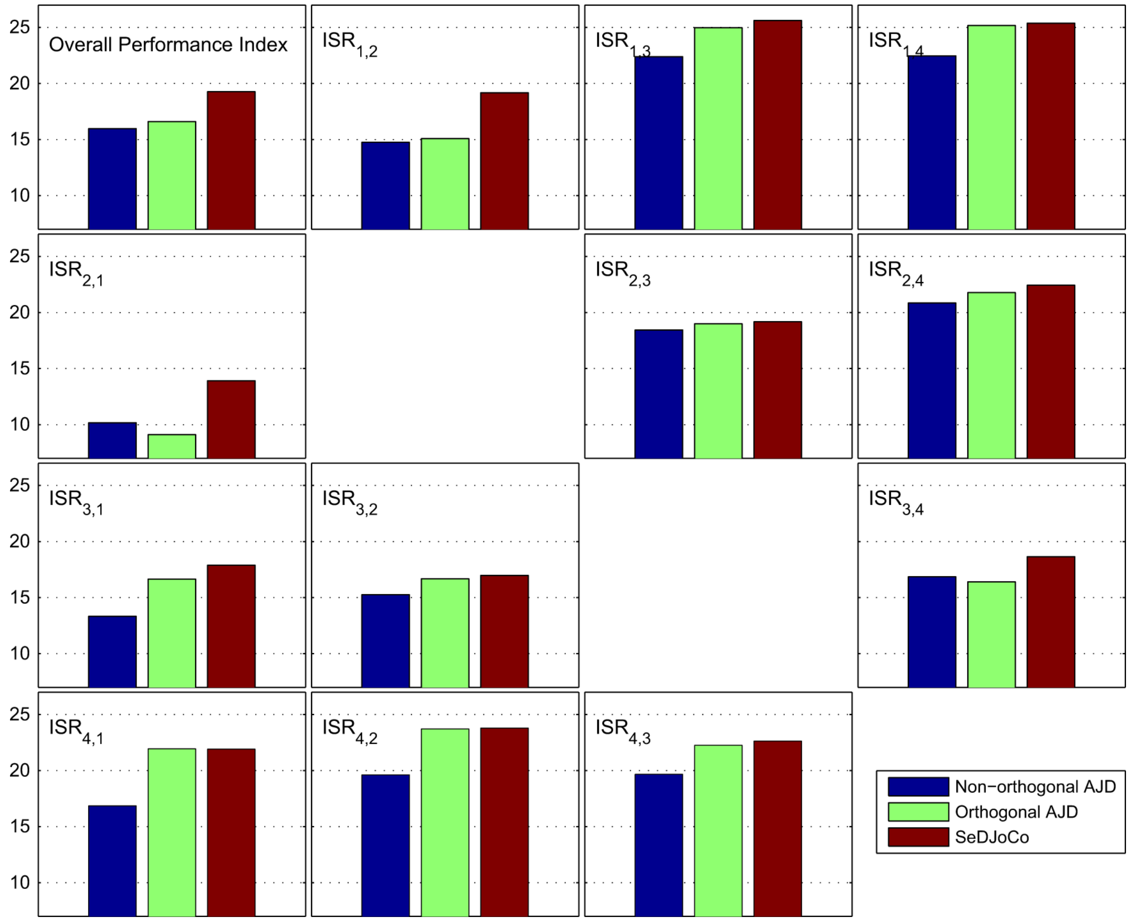


Fig. 4. Empirical mean ISR values and overall performance index (all in inverse dB). In each subplot the left, middle, and right bars show (respectively) the result of nonorthogonal AJD, orthogonal AJD, and HEAD.

much larger number of total receive antennas than the number of transmit antennas.

C. BSS of Complex-Valued Stationary Sources

ML separation of real-valued stationary Gaussian sources has been addressed in [11]–[13]. However, although complex-valued BSS has been considered extensively in the literature (e.g., [28]–[30]), ML separation has (to the best of our knowledge) only been addressed for the case of sources with independent, identically distributed (i.i.d.) time-structure (based on the sources' non-Gaussianity and/or noncircularity [28]–[30]), but not for (Gaussian) sources with temporal-domain or frequency-domain diversity. Although the extension of ML estimation from the real-valued to the complex-valued case is not straightforward in general (especially for noncircular sources), for circular Gaussian sources the basic principles are generally maintained, since the statistical information is maintained in the ordinary correlation matrices.

Therefore, in this subsection we demonstrate the separation of complex-valued circular stationary Gaussian sources, via a straightforward extension of the results in [13] (originally derived for the real-valued case), exploiting the complex-valued NCG algorithm so as to outperform "ordinary" complex-valued AJD.

Our experiment consists of a mixture of $K = 4$ complex-Gaussian sources, which are all generated as third-order moving average (MA) sources, each with three spectral zeros (in the Z-plane) taken from a pool of four zeros: $z_1 = 0.4e^{j\frac{\pi}{3}}$, $z_2 = 0.5e^{j\frac{2\pi}{3}}$, $z_3 = 0.6e^{j\frac{3\pi}{3}} = -0.6$ and $z_4 = 0.7e^{j\frac{4\pi}{3}}$. The kth source was generated as

$$s_k[t] = w_k[t] + h_k[1]w_k[t-1] + h_k[2]w_k[t-2] \\ + h_k[3]w_k[t-3], \quad k = 1,2,3,4 \quad (40)$$

for $t = 1, \dots T$, where

$$H_k(z) = 1 + h_k[1]z^{-1} + h_k[2]z^{-2} \\ + h_k[3]z^{-3} = \prod_{\substack{\ell=1 \\ \ell \neq k}}^{4} \left(1 - z_\ell z^{-1}\right) \quad (41)$$

such that $h_k[0] (= 1), h_k[1], h_k[2], h_k[3]$ are the (evidently complex-valued) MA coefficients of the kth source (such that $H_k(z)$ has all the four zeros defined above, except for the kth zero). The driving noise processes were all mutually independent, white circular complex-valued Gaussian noise processes. All sources were normalized to have unit power.

We have simulated 250 independent trials according to (1). The elements of the 4 × 4 mixing matrix have been drawn independently at each trial from an i.i.d. circular complex Gaussian distribution. We have used an observation length of $T = 100$ samples. The four target-matrices have been generated in each trial as

$$\boldsymbol{Q}_n = \boldsymbol{X}\boldsymbol{C}_n^{-1}\boldsymbol{X}^H, \quad n = 1, 2, 3, 4 \tag{42}$$

where $\boldsymbol{X} \triangleq [\boldsymbol{x}[1], \ldots, \boldsymbol{x}[T]]$ is an $N \times T = 4 \times 100$ matrix containing all the observed mixtures, and where $\boldsymbol{C}_n \triangleq E[\boldsymbol{s}_n\boldsymbol{s}_n^H]$ is the covariance matrix of the nth source, $\boldsymbol{s}_n \triangleq [s_n[1], \ldots, s_n[T]]^T$, $n = 1, 2, 3, 4$. The covariance matrices are assumed known (hence we work in a semiblind scenario), and are actually band-Toeplitz matrices, which are easily deduced from the MA structure of each source. These target-matrices are the complex-valued version of the target-matrices used for ML estimation in the real-valued case (see [13]). We present separation results obtained from applying to these target-matrices:

1) Ordinary nonorthogonal joint diagonalization (using the AC/DC algorithm [3]);
2) Ordinary orthogonal joint diagonalization (using [1]); and
3) SeDJoCo (using NCG, which is guaranteed to attain the ML estimate in this case, via its exact solution of SeDJoCo—unlike STJOCO, which might converge to a false solution).

In Fig. 4 we present the resulting mean interference-to–source ratio (ISR), defined in the form of an $N \times N$ matrix, whose (m, n)th element (denoted $ISR_{m,n}$) is the averaged square absolute value of the (m, n)th element (normalized by the (m, m)th element) of the overall mixing-demixing matrix $\boldsymbol{M} \triangleq \hat{\boldsymbol{B}}\boldsymbol{A}$ (where $\hat{\boldsymbol{B}}$ is the estimated separation matrix and $\boldsymbol{A}$ is the true mixing matrix). The values are shown in the figure in "inverse dB" values, showing each $-10\log_{10}(ISR_{m,n})$ $(m \neq n)$ in the (m, n)th subplot. In addition, we show in the $(1, 1)$ (upper left) subplot the averaged overall "performance-index" (also in "inverse dB"), which in our case (of equal-power sources with no scaling ambiguity) reduces to the mean of all ISRs.

The advantage of SeDJoCo over ordinary AJD for these target-matrices is evident in this example (for these ISR values). It is important to stress, however, that generally SeDJoCo cannot compete as a general AJD tool for other popular sets of target-matrices (e.g., empirical correlation matrices at different lags, as in SOBI [31])—its advantages in the BSS context are restricted to the specific sets of target-matrices used in the context of ML.

V. Conclusion

We have presented a specially structured AJD problem, which, given a set of N symmetric (in $\mathbb{R}^{N\times N}$) or Hermitian (in $\mathbb{C}^{N\times N}$) target-matrices $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N$, seeks a "sequentially drilled" joint congruence transformation thereof. More explicitly, the SeDJoCo problem aims to find a matrix $\boldsymbol{B}$, such that each transformed matrix $\boldsymbol{B}\boldsymbol{Q}_n\boldsymbol{B}^H$ $(n = 1, 2, \ldots, N)$ has all-zeros off-diagonal elements in its nth row and column. Thus, the transformed matrices exhibit a "sequential drilling" pattern, as illustrated in Fig. 1.

We have shown that the SeDJoCo problem arises in (at least) two different contexts of practical interest: One is maximum likelihood source separation (based on second-order statistics), and the other is coordinated beamforming for multiuser MIMO communications systems. We have also shown several alternative formulations of SeDJoCo, each providing a different perspective on the problem, and have proven that a solution always exists if all the target-matrices are positive-definite.

We have proposed two iterative solutions: One (termed NCG) is bases on Newton-type iterations, conveniently employing a conjugate-gradient solution if desired; and the other one (termed STJOCO) is based on a modification of an existing AJD algorithm employing LU decompositions. Upon convergence, NCG is guaranteed to yield an exact solution of SeDJoCo—whereas STJOCO may converge to local nonzero minima of its cost-function, which are not associated with an exact solution. However, in the context of CBF the STJOCO solutions tend to yield significantly stronger diagonal terms in the respective "drilled" columns of the target-matrices, thereby attaining better overall SINRs. The performance of both algorithms in both the CBF and complex-BSS contexts has been demonstrated and compared in several simulation experiments.

References

[1] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 161–164, 1996.

[2] A.-J. van der Veen, "Joint diagonalization via subspace fitting techniques," in *Proc. IEEE ICASSP'01*, 2001.

[3] A. Yeredor, "Nonorthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1545–1553, 2002.

[4] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller, "A fast algorithm for joint diagonalization with nonorthogonal transformations and its application to blind source separation," *J. Mach. Learn. Res.*, vol. 5, pp. 777–800, 2004.

[5] R. Vollgraf and K. Obermayer, "Quadratic optimization for simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3270–3278, 2006.

[6] B. Afsari, "Simple LU and QR based nonorthogonal matrix joint diagonalization," *Proc. ICA'06 Lecture Notes in Comput. Sci. (LNCS 3889)*, pp. 17–24, 2006.

[7] X.-L. Li and X. D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solutions," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1803–1814, 2007.

[8] P. Tichavský and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878–891, 2009.

[9] B. Song, F. Roemer, and M. Haardt, "Using a new structured joint congruence (STJOCO) transformation of Hermitian matrices for precoding in multiuser MIMO systems," in *Proc. IEEE ICASSP'10*, Dallas, TX, 2010.

[10] A. Yeredor, "On hybrid exact-approximate joint diagonalization," in *Proc. CAMSAP '09*, Aruba, Dutch Antilles, 2009.

[11] D.-T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1712–1725, 1997.

[12] S. Dégerine and A. Zaïdi, "Separation of an instantaneous mixture of Gaussian autoregressive sources by the exact maximum likelihood approach," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1492–1512, 2004.

[13] A. Yeredor, "Blind separation of Gaussian sources with general covariance structures: Bounds and optimal estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5057–5068, 2010.

[14] *Handbook of Blind Source Separation*, P. Comon and C. Jutten, Eds. New York: Academic, 2010.

[15] A. Yeredor, "Blind separation of Gaussian sources via second-order statistics with asymptotically optimal weighting," *IEEE Signal Process. Lett.*, vol. 7, no. 7, pp. 197–200, 2000.

[16] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, pp. 461–471, Feb. 2004.

[17] C.-B. Chae, D. Marzzarese, T. Inoue, and R. W. Heath, Jr, "Coordinated beamforming for the multiuser MIMO broadcast channel with limited feedforward," *IEEE Trans. Signal Process.*, vol. 56, pp. 6044–6056, 2008.

[18] Z. Pan, K.-K. Wong, and T.-S. Ng, "Generalized multiuser orthogonal space-division multiplexing," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 1969–1973, Nov. 2004.

[19] C.-B. Chae, D. Mazzarese, N. Jindal, and R. W. Heath, Jr, "Coordinated beamforming with limited feedback in MIMO broadcast channel," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 1505–1515, 2008.

[20] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc. Pts. F and H*, vol. 130, no. 1, pp. 11–16, 1983.

[21] A. van den Bos, "Complex gradient and Hessian," *IEE Proc.—Vision, Image Signal Process.*, vol. 141, no. 6, pp. 380–383, 1994.

[22] A. Greenbaum*, Iterative Methods for Solving Linear Systems, ser. Frontiers in Applied Mathematics*. Philadelphia, PA: SIAM, 1997.

[23] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. 25, pp. 772–781, 1978.

[24] B. Afsari, "Simple LU and QR based nonorthogonal matrix joint diagonalization," in *ICA 2006*, 2006, pp. 1–7.

[25] S. Boyd and L. Vandenberghe*, Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[26] V. Stankovic and M. Haardt, "Generalized design of multiuser MIMO precoding matrices," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 953–961, 2008.

[27] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian MIMO broadcast channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2004.

[28] J. Eriksson and V. Koivunen, "Complex random vectors and ICA models: Identifiability, uniqueness, and separability information theory," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1017–1029, 2006.

[29] P. Schreier, T. Adali, and L. Scharf, "On ICA of improper and noncircular sources," in *Proc. IEEE ICASSP'09*, 2009, pp. 3561–3564.

[30] L. Hualiang and T. Adali, "Stability analysis of complex maximum likelihood ICA using Wirtinger calculus," in *Proc. IEEE ICASSP'08*, Las Vegas, NV, 2008.

[31] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, 1997.

Arie Yeredor (M'99–SM'02) received the B.Sc. (*summa cum laude*) and Ph.D. degrees in electrical engineering from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1984 and 1997, respectively.

He is currently an Associate Professor with the School of Electrical Engineering, Department of Electrical Engineering-Systems, TAU, where his research and teaching areas are in statistical and digital signal processing. He also holds a consulting position with NICE Systems, Inc., Ra'anana, Israel, in the fields of speech and audio processing, video processing, and emitter location algorithms.

Prof. Yeredor previously served as an Associate Editor for the IEEE Signal Processing Letters and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He served as Technical Co-Chair of The Third International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP'2009) and more recently as General Co-Chair of the 10th International Conference on Latent Variables Analysis and Signal Separation (LVA/ICA2012). He has been awarded the yearly Best Lecturer of the Faculty of Engineering Award (at TAU) seven times. He served as a member of the IEEE Signal Processing Society's Signal Processing Theory and Methods (SPTM) Technical Committee in 2005–2010, and currently serves as Chair of the Signal Processing chapter of IEEE Israel Section.

Bin Song (S'08) was born in China on December 1, 1977. She received the Bachelor's degree in electronic engineering from the Zheng Zhou University, Zheng Zhou, China, in 1999, and the Master's degree in electrical and electronic engineering from University of Duisburg-Essen, Germany, in 2007.

Since March 2007, she has been a Research Assistant in the Communications Research Laboratory, Ilmenau University of Technology. Her current research interests include linear transmit-receive strategies for multiuser MIMO system and multidimensional signal processing.

Florian Roemer (S'04) has studied computer engineering at Ilmenau University of Technology, Germany, and McMaster University, Canada. He received the Diplom-Ingenieur (M.S.) degree with a major in communications engineering in October 2006. For his diploma thesis he received the Siemens Communications Academic Award 2006.

Since December 2006, he has been a Research Assistant in the Communications Research Laboratory, Ilmenau University of Technology. His research interests include multidimensional signal processing, high-resolution parameter estimation, as well as multiuser MIMO precoding and relaying.

Martin Haardt (S'90–M'98–SM'99) studied electrical engineering at Ruhr-University Bochum, Germany, and at Purdue University, West Lafayette, IN, receiving the Diplom-Ingenieur (M.S.) degree from Ruhr-University Bochum in 1991 and the Doktor-Ingenieur (Ph.D.) degree from Munich University of Technology, Germany, in 1996.

In 1997, he joined Siemens Mobile Networks, Munich, Germany, where he was responsible for strategic research for third generation mobile radio systems. From 1998 to 2001, he was the Director for International Projects and University Cooperations in the mobile infrastructure business of Siemens in Munich, where his work focused on mobile communications beyond the third generation. During his time at Siemens, he also taught in the international Master of Science in Communications Engineering program at Munich University of Technology. He has been a Full Professor in the Department of Electrical Engineering and Information Technology and Head of the Communications Research Laboratory at Ilmenau University of Technology, Germany, since 2001. In fall 2006 and fall 2007, he was a visiting professor at the University of Nice in Sophia-Antipolis, France, and at the University of York, UK, respectively. His research interests include wireless communications, array signal processing, high-resolution parameter estimation, as well as numerical linear and multi-linear algebra.

Dr. Haardt has received the 2009 Best Paper Award from the IEEE Signal Processing Society, the Vodafone (formerly Mannesmann Mobilfunk) Innovations-Award for outstanding research in mobile communications, the ITG best paper award from the Association of Electrical Engineering, Electronics, and Information Technology (VDE), and the Rohde & Schwarz Outstanding Dissertation Award. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2002–2006 and since 2011), the IEEE SIGNAL PROCESSING LETTERS (2006–2010), the RESEARCH LETTERS IN SIGNAL PROCESSING (2007–2009), the *Hindawi Journal of Electrical and Computer Engineering* (since 2009), the *EURASIP Signal Processing Journal* (since 2011), and as a Guest Editor for the *EURASIP Journal on Wireless Communications and Networking*. He has also served as an elected member of the Sensor Array and Multichannel (SAM) Technical Committee of the IEEE Signal Processing Society (since 2011), as the Technical Co-Chair of the IEEE International Symposiums on Personal Indoor and Mobile Radio Communications (PIMRC) 2005 in Berlin, Germany, and as the Technical Program Chair of the IEEE International Symposium on Wireless Communication Systems (ISWCS) 2010 in York, U.K.