4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan

A GENERALIZATION OF A CLASS OF BLIND SOURCE SEPARATION ALGORITHMS FOR CONVOLUTIVE MIXTURES

Herbert Buchner, Robert Aichner, Walter Kellermann

Multimedia Communications and Signal Processing,
University of Erlangen-Nuremberg
Cauerstr. 7, D-91058 Erlangen, Germany
{buchner,aichner,wk}@LNT.de

ABSTRACT

There are two main approaches for blind source separation (BSS) on time series using second-order statistics. One is to utilize the nonwhiteness property, and the other one is to utilize the nonstationarity property of the source signal. In this paper, we combine both approaches for convolutive mixtures using a matrix notation that leads to a number of new insights. We give rigorous derivations of the corresponding time-domain and frequency-domain approaches by generalizing a known cost function so that it inherently allows joint optimization for several time lags of the correlations. The approach is suitable for on-line and off-line algorithms by introducing a general weighting function allowing for tracking of time-varying environments. For both, the time-domain and frequency-domain versions, we discuss links to well-known and also to extended algorithms as special cases. Moreover, using the so-called generalized coherence, we establish links between the time-domain and frequency-domain algorithms and show that our cost function leads to an update equation with an inherent normalization.

1. INTRODUCTION

The problem of separating convolutive mixtures of unknown time series arises in several application domains, a prominent example being the so-called cocktail party problem, where we want to recover the speech signals of multiple speakers who are simultaneously talking in a room. The room may be very reverberant due to reflections on the walls, i.e., the original source signals $s_q(n)$, $q = 1, \ldots, Q$ of our separation problem are filtered by a multiple input and multiple output (MIMO) system before they are picked up by the sensors. In the following, we assume that the number Q

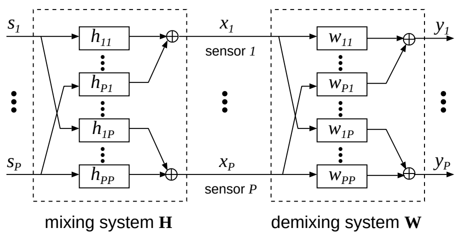

Fig. 1. Linear MIMO model for BSS.

of source signals $s_q(n)$ equals the number of sensor signals $x_p(n)$, $p = 1, \ldots, P$ (Fig. 1). An M-tap mixing system is thus described by

$$x_p(n) = \sum_{q=1}^{P} \sum_{\kappa=0}^{M-1} h_{qp}(\kappa) s_q(n-\kappa), \tag{1}$$

where $h_{qp}(\kappa)$, $\kappa = 0, \ldots, M-1$ denote the coefficients of the filter from the q-th source to the p-th sensor.

In BSS, we are interested in finding a corresponding demixing system according to Fig. 1, where the output signals $y_q(n)$, $q = 1, \ldots, P$ are described by

$$y_q(n) = \sum_{p=1}^{P} \sum_{\kappa=0}^{L-1} w_{pq}(\kappa) x_p(n-\kappa). \tag{2}$$

It can be shown (see, e.g., [1]) that the MIMO demixing system coefficients $w_{pq}(\kappa)$ can in fact reconstruct the sources up to an unknown permutation and an unknown filtering of the individual signals, where L should be chosen at least equal to M.

In order to estimate the P^2L MIMO coefficients $w_{pq}(\kappa)$, we consider in this paper only approaches using *second-order statistics*. It has been shown that on real-world signals with some time-structure, second-order statistics generates enough constraints to solve the BSS problem in principle, by utilizing one of the following two signal properties [1]:

- Nonwhiteness property by simultaneous diagonalization of output correlation matrices over multiple time-lags, e.g., [2],
- Nonstationarity property by simultaneous diagonalization of short-time output correlation matrices at different time intervals, e.g., [3]-[8].

While there are several algorithms for convolutive mixtures utilizing nonstationarity, both in the time domain and in the frequency domain, there are currently very few approaches taking the nonwhiteness property into account. Although in theory, each of these properties is known to be sufficient, it has recently been shown that in practical scenarios, the combination of these criteria can lead to improved performance [9, 10].

In the following, we present a rigorous derivation of a more general class of algorithms for convolutive mixtures by first introducing a general matrix formulation for convolutive mixtures following [11] that includes all time lags. The approach utilizes both, the nonwhiteness property and the nonstationarity property and is suitable for on-line and off-line algorithms by introducing a general weighting function allowing for tracking of time-varying environments. For both, the time-domain and frequency-domain versions, we discuss links to well-known and extended algorithms as special cases. Moreover, using the so-called generalized coherence [12], we establish links between the time-domain and frequency-domain algorithms and show that our cost function leads to an update equation with an inherent normalization.

2. A GENERIC BLOCK TIME-DOMAIN BSS ALGORITHM

2.1. Matrix formulation for convolutive mixtures with extension to several time lags

From Fig. 1, it can be seen that the output signals $y_q(n)$, $(q = 1, \dots, P)$ of the unmixing system at time n are given by

$$y_q(n) = \sum_{p=1}^{P} \mathbf{x}_p^T(n)\mathbf{w}_{pq}, \tag{3}$$

where

$$\mathbf{x}_p(n) = [x_p(n), x_p(n-1), \dots, x_p(n-L+1)]^T$$

is a vector containing the latest L samples of the sensor signal x_p of the p-th channel, and where

$$\mathbf{w}_{pq} = [w_{pq,0}, w_{pq,1}, \dots, w_{pq,L-1}]^T$$

contains the current weights of the MIMO filter taps from the p-th sensor channel to the q-th output channel. Superscript ^{T} denotes transposition of a vector or a matrix.

We now define the corresponding block output signal vector. To simplify the presentation, we consider a block length that is equal to the filter length L in this paper. From (3) follows

$$\mathbf{y}_q(m) = \sum_{p=1}^{P} \mathbf{U}_p^T(m)\mathbf{w}_{pq}, \tag{4}$$

with m being the block time index, and

$$\begin{aligned} \mathbf{y}_q(m) &= [y_q(mL), \dots, y_q(mL+L-1)]^T, && (5)\\ \mathbf{U}_p(m) &= [\mathbf{x}_p(mL), \dots, \mathbf{x}_p(mL+L-1)]. && (6) \end{aligned}$$

It can be verified that $\mathbf{U}_p$, $p = 1, \dots, P$ are Toeplitz matrices of size $(L \times L)$:

$$\mathbf{U}_p^T(m) = \begin{bmatrix} x_p(mL) & \cdots & x_p(mL-L+1) \\ x_p(mL+1) & \ddots & x_p(mL-L+2) \\ \vdots & \ddots & \vdots \\ x_p(mL+L-1) & \cdots & x_p(mL) \end{bmatrix}.$$

Next, in order to rigorously introduce multiple *time lags* in the cost function below, we now extend the output signal vector (5) to the following $L \times L$ matrix by incremental shifts of each column by one sample:

$$\mathbf{Y}_q(m) = \begin{bmatrix} y_q(mL) & \cdots & y_q(mL-L+1) \\ y_q(mL+1) & \ddots & y_q(mL-L+2) \\ \vdots & \ddots & \vdots \\ y_q(mL+L-1) & \cdots & y_q(mL) \end{bmatrix}.$$

Using this definition, (4) becomes

$$\mathbf{Y}_q(m) = \sum_{p=1}^{P} \mathbf{X}_p(m)\mathbf{W}_{pq}, \tag{7}$$

where the $L \times 2L$ matrices $\mathbf{X}_p(m)$ are obtained from the Toeplitz matrices $\mathbf{U}_p$ by doubling their size, i.e.,

$$\mathbf{X}_p(m) = [\mathbf{U}_p^T(m),\ \mathbf{U}_p^T(m-1)]. \tag{8}$$

The matrices $\mathbf{U}_p^T(m-1)$ are also Toeplitz so that the first row of $\mathbf{X}_p(m)$ contains $2L$ input samples and each subsequent row is shifted to the right by one sample and thus contains one new input sample. $\mathbf{W}_{pq}$ are $2L \times L$ Sylvester matrices, which are defined as

$$\mathbf{W}_{pq}(m) = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{pq,0} & \ddots & \vdots \\ \vdots & w_{pq,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{pq,0} \\ 0 & w_{pq,L-1} & \ddots & w_{pq,1} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{pq,L-1} \end{bmatrix}. \tag{9}$$

Finally, to allow a convenient notation of the algorithm combining all channels, we write (7) compactly as

$$\mathbf{Y}(m) = \mathbf{X}(m)\mathbf{W}, \tag{10}$$

with the matrices

$$\begin{aligned} \mathbf{Y}(m) &= [\mathbf{Y}_1(m), \cdots, \mathbf{Y}_P(m)], && (11)\\ \mathbf{X}(m) &= [\mathbf{X}_1(m), \cdots, \mathbf{X}_P(m)], && (12)\\ \mathbf{W} &= \begin{bmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{P1} & \cdots & \mathbf{W}_{PP} \end{bmatrix}. && (13) \end{aligned}$$

2.2. Cost function and algorithm derivation

Having defined the compact matrix formulation (10) for the block-MIMO filtering, we now define the following cost function as a generalization of [5]:

$$\begin{aligned} \mathcal{J}(m) = \sum_{i=0}^{m} \beta(i,m) \Big\{ &\log\det \operatorname{bdiag} \mathbf{Y}^H(i)\mathbf{Y}(i) \\ &- \log\det \mathbf{Y}^H(i)\mathbf{Y}(i) \Big\}, \end{aligned} \tag{14}$$

where β is a window function that is normalized according to $\sum_{i=0}^{m} \beta(i,m) = 1$ which allows off-line and on-line implementations of the algorihms (e.g., $\beta(i,m) = (1-\lambda)\lambda^{m-i}$ leads to an efficient on-line version allowing for tracking in time-varying environments). Since we use the matrix formulation (10) for calculating the short-time correlation matrices $\mathbf{Y}^H(m)\mathbf{Y}(m)$, the cost function inherently includes all time-lags of all auto-correlations and cross-correlations of the BSS output signals. The bdiag operation on a partitioned block matrix consisting of several submatrices sets all submatrices on the off-diagonals to zero. In our case, the block matrices refer to the different signal channels. By Oppenheim's inequality [13], it is ensured that the first term in the braces in (14) is always greater than or equal to the second term, where the equality is only valid if all block-offdiagonal elements of $\mathbf{Y}^H\mathbf{Y}$, i.e., the *output cross-correlations over all time-lags*, vanish. In geometrical terms this can be interpreted as a simultaneous orthogonalization relative to several subspaces, since the determinant of a matrix corresponds to a volume of a parallelepiped spanned by the column vectors (Fig. 2).

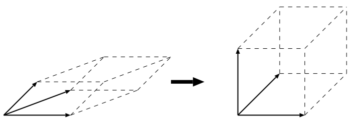

Fig. 2. Parallelepiped.

For the following derivations, we omit the block-time index m for simplicity, and first define the short-time correlation matrices

$$\mathbf{R}_{xx} = \mathbf{X}^H\mathbf{X}, \quad (15)$$
$$\mathbf{R}_{yy} = \mathbf{Y}^H\mathbf{Y}. \quad (16)$$

Note that in principle, there are two basic methods to estimate the output correlation matrices for nonstationary output signals: the so-called correlation method, and the covariance method as they are known from linear prediction problems [14]. While the correlation method leads to a slightly lower computational complexity (and to smaller matrices, when implemented in the frequency domain covered in Section 3), we consider the more accurate covariance method in this paper.

For the derivation of the gradient

$$\nabla_{\mathbf{W}}\mathcal{J}(m) = 2\frac{\partial \mathcal{J}(m)}{\partial \mathbf{W}^*},$$

we use the expressions (e.g., [15])

$$\frac{\partial}{\partial \mathbf{W}^*}\log\det \mathbf{W}^H\mathbf{R}_{xx}\mathbf{W} = 2\mathbf{R}_{xx}\mathbf{W}(\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W})^{-1}$$

and

$$\frac{\partial}{\partial \mathbf{W}^*}\log\det \operatorname{bdiag}\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W} = 2\mathbf{R}_{xx}\mathbf{W}(\operatorname{bdiag}\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W})^{-1}.$$

Using these relations, it follows from (14)

$$\begin{aligned}\nabla_{\mathbf{W}}\mathcal{J}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{R}_{xx}\mathbf{W}\left\{\operatorname{bdiag}^{-1}\mathbf{R}_{yy} - \mathbf{R}_{yy}^{-1}\right\}\\ &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{R}_{xy}\left\{\operatorname{bdiag}^{-1}\mathbf{R}_{yy} - \mathbf{R}_{yy}^{-1}\right\}\\ &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\left\{\mathbf{R}_{yy} - \operatorname{bdiag}\mathbf{R}_{yy}\right\}\\ &\quad \cdot\operatorname{bdiag}^{-1}\mathbf{R}_{yy}. \end{aligned} \quad (17)$$

2.3. Equivariance property and natural gradient

With an iterative optimization procedure, the current demixing matrix is obtained by the recursive update equation

$$\mathbf{W}(m) = \mathbf{W}(m-1) - \mu\Delta\mathbf{W}(m), \quad (18)$$

where μ is a stepsize parameter, and $\Delta\mathbf{W}(m)$ is the update which is set equal to $\nabla_{\mathbf{W}}\mathcal{J}(m)$ for gradient descent adaptation. However, it is known that stochastic gradient descent suffers from slow convergence in many practical problems due to dependencies in the data being processed.

In the BSS application, we can show that the separation performance using (18) together with (17) depends on the MIMO mixing system. The mixing process can be described analogously to (10) by $\mathbf{X} = \mathbf{SH}$, where $\mathbf{S}$ is the corresponding $L \times PL$ source signal matrix with time shifts, and $\mathbf{H}$ is the $PL \times 2PL$ mixing matrix in Sylvester structure. Due to the inevitable filtering ambiguity in convolutive BSS, it is at best possible to obtain an arbitrary *block diagonal* matrix $\mathbf{C} = \mathbf{HW}$. To see how (17) behaves, we pre-multiply both sides of (17) by $\mathbf{H}$. This way it can easily be shown that $\mathbf{C}(m)$ depends on the mixing system $\mathbf{H}$.

Fortunately, a modification of the ordinary gradient has been developed that largely removes all effects of an ill-conditioned mixing matrix $\mathbf{H}$. Termed the *natural gradient* by Amari [16] and the *relative gradient* by Cardoso [17], this modification is usually written in the following way:

$$\Delta\mathbf{W} = \nabla_{\mathbf{W}}\mathcal{J}\mathbf{W}^H\mathbf{W}.$$

For our approach, we have to slightly modify it to

$$\Delta\mathbf{W} = \mathbf{W}\mathbf{W}^H\nabla_{\mathbf{W}}\mathcal{J},$$

which finally leads to the following update:

$$\begin{aligned}\Delta\mathbf{W}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{W}\left\{\mathbf{R}_{yy} - \operatorname{bdiag}\mathbf{R}_{yy}\right\}\\ &\quad \cdot\operatorname{bdiag}^{-1}\mathbf{R}_{yy}. \end{aligned} \quad (19)$$

To see that the above formulation of the natural gradient is justified, we again pre-multiply the update (19), which leads to

$$\Delta\mathbf{C}(m) = 4\sum_{i=0}^{m}\beta(i,m)\mathbf{C}\left\{\mathbf{R}_{yy} - \operatorname{bdiag}\mathbf{R}_{yy}\right\}\operatorname{bdiag}^{-1}\mathbf{R}_{yy}.$$

The evolutionary behaviour of $\mathbf{C} = \mathbf{C}(m)$ depends only on the estimated source signal vector sequence and μ, and the mixing matrix $\mathbf{H}$ has been absorbed as an initial condition into $\mathbf{C}(0) = \mathbf{HW}(0)$ as desired. The uniform performance provided by (19) is due to the so-called equivariance property provided by the natural/relative gradient BSS update [17]. In our case, only the modified relative gradient exhibits this property.

2.4. Special cases and links to known time-domain algorithms

To analyze the generalized update (19), and to study links to some known algorithms, we consider now the case $P = 2$ for simplicity. In this case, we have

$$\begin{aligned}\Delta\mathbf{W}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{W}\\ &\quad \cdot\begin{bmatrix}\mathbf{0} & \mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1}\\ \mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{0}\end{bmatrix}\\ &= 4\sum_{i=0}^{m}\beta(i,m)\\ &\quad \cdot\begin{bmatrix}\mathbf{W}_{12}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{11}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1}\\ \mathbf{W}_{22}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{21}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1}\end{bmatrix}, \end{aligned} \quad (20)$$

where $\mathbf{R}_{y_p y_q}$, $p, q \in \{1, 2\}$ are the corresponding submatrices of $\mathbf{R}_{yy}$.

In [9, 10], a time-domain algorithm was presented that copes very well with reverberant acoustic environments. Although it was originally introduced as a heuristic extension of [5] incorporating several time lags, this algorithm can be directly obtained from (19) or (20) by approximating the block diagonals of $\mathbf{R}_{yy}(m)$ by the output signal powers, i.e.,

$$\tilde{\mathbf{R}}_{y_q y_q}(m) = \operatorname{diag} \mathbf{R}_{y_q y_q}(m) = \mathbf{y}_q^H(m)\mathbf{y}_q(m)\mathbf{I}$$

for $q = 1, \ldots, P$. Using this approximation, the remaining products of Sylvester matrices and Toeplitz matrices in the update equation (20) can be efficiently implemented by a (fast) convolution as was done in [10].

If we do not apply this approximation, we can slightly simplify the update equation (20) in a different way by considering (16) and noting that $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ for any nonsingular square matrices $\mathbf{A}$ and $\mathbf{B}$:

$$\Delta \mathbf{W}(m) = 4 \sum_{i=0}^{m} \beta(i, m) \mathbf{W} \begin{bmatrix} \mathbf{0} & \mathbf{Y}_1^H \mathbf{Y}_2^{-H} \\ \mathbf{Y}_2^H \mathbf{Y}_1^{-H} & \mathbf{0} \end{bmatrix},$$

where ${}^{-H}$ denotes conjugate transposition of an inverse matrix. This formulation not only reduces the complexity but the matrices to be inverted are also much better conditioned, since the condition number of $\mathbf{Y}^H\mathbf{Y}$ in (16) is the square of the condition number of $\mathbf{Y}$ [18].

3. GENERIC FREQUENCY-DOMAIN BSS

Frequency-domain BSS is very popular for convolutive BSS since all techniques originally developed for instantaneous BSS can be applied independently in each frequency bin, e.g., [1, 6, 7, 8]. Unfortunately, the permutation problem, which is inherent in BSS, may then also appear independently in each frequency bin so that extra measures have to be taken to avoid this internal permutation. Based on the above matrix formulation in the time domain, the following derivation of frequency-domain algorithms shows explicitly the relation between time-domain and frequency-domain algorithms, as well as some extensions. Moreover, from a link with [8], it becomes clear that (14) leads to the very desirable property of an inherent stepsize normalization.

3.1. General frequency-domain formulation

The matrix formulation introduced above for the time-domain allows a rigorous derivation of the corresponding frequency-domain BSS algorithms. In the frequency domain, the structure of the algorithm depends on the method chosen for estimating the correlation matrices. Here, we consider again the more accurate covariance method [14]. The matrices $\mathbf{X}_p(m)$ and $\mathbf{W}_{pq}$ are now diagonalized in two steps. We first consider the $L \times 2L$ Toeplitz matrices $\mathbf{X}_p(m)$

Step 1: Transformation of Toeplitz matrices into circulant matrices.
Any Toeplitz matrix $\mathbf{X}_p$ can be transformed, by doubling its size, to a circulant matrix $\mathbf{C}_{X_p}(m)$ [11]. In our case we define the circulant matrix by taking into account (8) by

$$\mathbf{C}_{X_p}(m) = \begin{bmatrix} \mathbf{X}_p'(m-3) & \mathbf{X}_p(m-1) \\ \mathbf{X}_p(m-2) & \mathbf{X}_p(m) \\ \mathbf{X}_p(m-1) & \mathbf{X}_p'(m-3) \\ \mathbf{X}_p(m) & \mathbf{X}_p(m-2) \end{bmatrix},$$

where $\mathbf{X}_p'(m-3) = [\mathbf{U}_p^T(m-3), \mathbf{U}_p^T(m)]$. It follows

$$\mathbf{X}_p(m) = \mathbf{W}_{L\times 4L}^{0001} \mathbf{C}_{X_p}(m) \mathbf{W}_{4L\times 2L}^{10}, \tag{21}$$

where we introduced the windowing matrices

$$\begin{aligned} \mathbf{W}_{L\times 4L}^{0001} &= [\mathbf{0}_{L\times L}, \mathbf{0}_{L\times L}, \mathbf{0}_{L\times L}, \mathbf{I}_{L\times L}], \\ \mathbf{W}_{4L\times 2L}^{10} &= [\mathbf{I}_{2L\times 2L}, \mathbf{0}_{2L\times 2L}]^T. \end{aligned}$$

Step 2: Transformation of the circulant matrices into diagonal matrices.
Using the $4L \times 4L$ DFT matrix $\mathbf{F}_{4L\times 4L}$, the circulant matrices are diagonalized as follows:

$$\mathbf{C}_{X_p}(m) = \mathbf{F}_{4L\times 4L}^{-1} \underline{\mathbf{X}}_p(m) \mathbf{F}_{4L\times 4L},$$

where the diagonal matrices $\underline{\mathbf{X}}_p(m)$ can be expressed by the first columns of $\mathbf{C}_{X_p}(m)$,

$$\begin{aligned} &\underline{\mathbf{X}}_p(m) = \\ &\quad \operatorname{diag}\{\mathbf{F}_{4L\times 4L}[x_p(mL-3L), \ldots, x_p(mL-1), \\ &\quad x_p(mL), x_p(mL+1), \ldots, x_p(mL+L-1)]^T\}, \end{aligned} \tag{22}$$

i.e., to obtain $\underline{\mathbf{X}}_p(m)$, we transform the concatenated vectors of the current block and three previous blocks of the input signals $x_p(n)$.

Now, (21) can be rewritten equivalently as

$$\mathbf{X}_p(m) = \mathbf{W}_{L\times 4L}^{0001} \mathbf{F}_{4L\times 4L}^{-1} \underline{\mathbf{X}}_p(m) \mathbf{F}_{4L\times 4L} \mathbf{W}_{4L\times 2L}^{10}, \tag{23}$$

Equations (23) and (22) exhibit a form that is structurally similar to that of the corresponding counterparts of the well-known (supervised) frequency-domain adaptive filters [11]. However, the major difference here is that we need a transformation length of at least $4L$ instead of $2L$. This should come as no surprise, since in BSS using the covariance method, both convolution and correlation is carried out where both operations double the transformation length.

We now transform the matrices $\mathbf{W}_{pq}$ in the same way as shown above for $\mathbf{X}_p$. Thereby, we obtain

$$\mathbf{W}_{pq} = \mathbf{W}_{2L\times 4L}^{10} \mathbf{F}_{4L\times 4L}^{-1} \underline{\mathbf{W}}_{pq} \mathbf{F}_{4L\times 4L} \mathbf{W}_{4L\times L}^{1000}, \tag{24}$$

where

$$\begin{aligned} \mathbf{W}_{4L\times L}^{1000} &= [\mathbf{I}_{L\times L}, \mathbf{0}_{L\times L}, \mathbf{0}_{L\times L}, \mathbf{0}_{L\times L}]^T, \\ \mathbf{W}_{2L\times 4L}^{10} &= [\mathbf{I}_{2L\times 2L}, \mathbf{0}_{2L\times 2L}] = \left(\mathbf{W}_{4L\times 2L}^{10}\right)^T, \end{aligned}$$

and

$$\underline{\mathbf{W}}_{pq} = \operatorname{diag}\{\mathbf{F}_{4L\times 4L}[w_{pq,0}, \ldots, w_{pq,L-1}, 0, \ldots, 0]^T\}.$$

Note that in view of the structure of the matrix $\mathbf{W}_{pq}$, i.e., the shifted columns in (9), it can be shown that the pre-multiplied transformation $\mathbf{W}_{2L\times 4L}^{10} \mathbf{F}_{4L\times 4L}^{-1}$ in (24) is related to the demixing filter taps in the first column of $\mathbf{W}_{pq}$, while the post-multiplied transformation in (24), which we denote by

$$\mathbf{L}_{4L\times L}^{1000} = \mathbf{F}_{4L\times 4L} \mathbf{W}_{4L\times L}^{1000},$$

is related to the introduction of the multiple time-lags (see also Section 3.2). Combining all channels, we obtain

$$\begin{aligned}\mathbf{X}(m) &= \mathbf{W}^{0001}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{X}}(m) \\ &\quad\cdot \mathrm{bdiag}\{\mathbf{F}_{4L\times 4L}\mathbf{W}^{10}_{4L\times 2L},\dots \\ &\quad \dots,\mathbf{F}_{4L\times 4L}\mathbf{W}^{10}_{4L\times 2L}\}, &(25)\\ \mathbf{W} &= \mathrm{bdiag}\{\mathbf{W}^{10}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L},\dots \\ &\quad \dots,\mathbf{W}^{10}_{2L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\}\underline{\mathbf{W}}(m)\mathbf{L}, &(26)\end{aligned}$$

where $\mathrm{bdiag}\{\mathbf{A}_1,\dots,\mathbf{A}_P\}$ denotes a block-diagonal matrix with submatrices $\mathbf{A}_1,\dots,\mathbf{A}_P$ on its diagonal, $\underline{\mathbf{X}}(m)$ and $\underline{\mathbf{W}}(m)$ are defined analogously to (12) and (13), respectively. $\mathbf{L}$ denotes the $4LP \times LP$ matrix

$$\mathbf{L} = \mathrm{bdiag}\{\mathbf{L}^{1000}_{4L\times L},\dots,\mathbf{L}^{1000}_{4L\times L}\}.$$

From (10),(25), and (26) we further obtain

$$\mathbf{Y} = \mathbf{W}^{0001}_{L\times 4L}\mathbf{F}^{-1}_{4L\times 4L}\underline{\mathbf{X}}(m)\mathbf{G}^{10}_{4LP\times 4LP}\underline{\mathbf{W}}(m)\mathbf{L}, \qquad (27)$$

where

$$\begin{aligned}\mathbf{G}^{10}_{4LP\times 4LP} &= \mathrm{bdiag}\left\{\mathbf{G}^{10}_{4L\times 4L},\dots,\mathbf{G}^{10}_{4L\times 4L}\right\},\\ \mathbf{G}^{10}_{4L\times 4L} &= \mathbf{F}_{4L\times 4L}\mathbf{W}^{10}_{4L\times 4L}\mathbf{F}^{-1}_{4L\times 4L},\\ \mathbf{W}^{10}_{4L\times 4L} &= \mathbf{W}^{10}_{4L\times 2L}\mathbf{W}^{10}_{2L\times 4L}\\ &= \begin{bmatrix}\mathbf{I}_{2L\times 2L} & \mathbf{0}_{2L\times 2L}\\ \mathbf{0}_{2L\times 2L} & \mathbf{0}_{2L\times 2L}\end{bmatrix}.\end{aligned}$$

To formulate the cost function (14) equivalently in the frequency domain, we now need to calculate the short-time correlation matrix using (27), i.e.,

$$\begin{aligned}\mathbf{R}_{yy} &= \mathbf{Y}^H\mathbf{Y}\\ &= \mathbf{L}^H\mathbf{S}_{yy}\mathbf{L},\end{aligned}$$

where

$$\begin{aligned}\mathbf{S}_{yy} &= \underline{\mathbf{W}}^H\mathbf{S}_{xx}\underline{\mathbf{W}},\\ \mathbf{S}_{xx} &= \left(\mathbf{G}^{10}_{4LP\times 4LP}\right)^H\underline{\mathbf{X}}^H\mathbf{G}^{0001}_{4L\times 4L}\underline{\mathbf{X}}\\ &\quad\cdot\mathbf{G}^{10}_{4LP\times 4LP},\\ \mathbf{G}^{0001}_{4L\times 4L} &= \mathbf{F}_{4L\times 4L}\mathbf{W}^{0001}_{4L\times 4L}\mathbf{F}^{-1}_{4L\times 4L},\\ \mathbf{W}^{0001}_{4L\times 4L} &= \mathbf{W}^{0001}_{4L\times L}\mathbf{W}^{0001}_{L\times 4L}\\ &= \begin{bmatrix}\mathbf{0}_{3L\times 3L} & \mathbf{0}_{3L\times L}\\ \mathbf{0}_{L\times 3L} & \mathbf{I}_{L\times L}\end{bmatrix}.\end{aligned}$$

The derivation of the gradient of the cost function in the frequency domain is now done in a similar way as in the time domain. Since $\mathbf{Y}^H\mathbf{Y} = \mathbf{L}^H\underline{\mathbf{W}}^H\mathbf{S}_{xx}\underline{\mathbf{W}}\mathbf{L}$, we obtain using the chain rule $\nabla_{\underline{\mathbf{W}}}\mathcal{J} = \nabla_{(\underline{\mathbf{W}}\mathbf{L})}\mathcal{J}\cdot\mathbf{L}^H$, and thus

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}_{xy}\mathbf{L}\left\{\mathrm{bdiag}^{-1}\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L}\right.\\ &\quad\left.-\left(\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L}\right)^{-1}\right\}\mathbf{L}^H\\ &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}_{xy}\mathbf{L}(\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L})^{-1}\mathbf{L}^H\\ &\quad\cdot\{\mathbf{S}_{yy}-\mathrm{bdiag}\,\mathbf{S}_{yy}\}\,\mathbf{L}\\ &\quad\cdot\mathrm{bdiag}^{-1}(\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L})\mathbf{L}^H, &(28)\\ \mathbf{S}_{xy} &= \mathbf{S}_{xx}\underline{\mathbf{W}}. &(29)\end{aligned}$$

Although it is straightforward, we do not consider the natural gradient here in the frequency domain for simplicity.

3.2. The constraints and the permutation problem in frequency-domain BSS

Two types of constraints appear in the gradient (28):

- Matrix $\mathbf{L}$ introduces joint diagonalization over all time-lags
- The matrices $\mathbf{G}^{::}_{::}$ are mainly responsible for preventing the internal permutation among the different frequency bins.

Current frequency-domain BSS algorithms do not take the nonwhiteness property into account. By neglecting matrix $\mathbf{L}$ in (28) we obtain the simplified algorithm utilizing only the nonstationarity of the source signals:

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\\ &\quad\cdot\{\mathbf{S}_{yy}-\mathrm{bdiag}\,\mathbf{S}_{yy}\}\,\mathrm{bdiag}^{-1}\,\mathbf{S}_{yy}. &(30)\end{aligned}$$

Note that this algorithm still avoids the well-known internal permutation problem of frequency-domain BSS using the constraints $\mathbf{G}^{::}_{::}$ in $\mathbf{S}_{xy}$ and $\mathbf{S}_{yy}$.

By additionally removing these constraints, i.e., by approximating $\mathbf{G}^{::}_{::}$ as scaled identity matrices [11], all the submatrices in (30) become diagonal matrices. Only in this case (30) can be decomposed in its frequency components, i.e., we can equivalently write

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}^{(\nu)}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}^{(\nu)}_{xy}\left(\mathbf{S}^{(\nu)}_{yy}\right)^{-1}\\ &\quad\cdot\left\{\mathbf{S}^{(\nu)}_{yy}-\mathrm{diag}\,\mathbf{S}^{(\nu)}_{yy}\right\}\mathrm{diag}^{-1}\,\mathbf{S}^{(\nu)}_{yy}, &(31)\end{aligned}$$

where $\nu = 0,\dots,4L-1$ denotes the frequency bins. In contrast to $\mathbf{S}_{xy}$ and $\mathbf{S}_{yy}$ in (30) which are $4LP \times 4LP$ matrices each, the corresponding matrices $\mathbf{S}^{(\nu)}_{xy}$ and $\mathbf{S}^{(\nu)}_{yy}$ in (31) are only of dimension $P \times P$. While (31) is computationally more efficient than (30), the known measures to avoid the internal permutation have to be taken.

3.3. Links to known frequency-domain algorithms and the generalized coherence

The unconstrained coefficient update (31) is directly related to some known frequency-domain BSS algorithms. In [7], an algorithm that is similar to (31) was derived by directly optimizing a cost function similar to the one in [5] in a bin-wise manner. More recently, Fancourt and Parra proposed in [8] to apply the magnitude-squared coherence

$$|\gamma^{(\nu)}_{y_py_q}(m)|^2 = \frac{|S^{(\nu)}_{y_py_q}(m)|^2}{S^{(\nu)}_{y_py_p}(m)S^{(\nu)}_{y_qy_q}(m)}, \qquad (32)$$

$p,q \in \{1,\dots,P\}$ as a cost function for frequency-domain BSS, where $S^{(\nu)}_{y_py_q}(m)$ denotes the (p,q)-th element of $\mathbf{S}^{(\nu)}_{yy}(m)$, i.e., the power spectral density in the ν-th bin and block m. The coherence (32) has the very desirable property that

$$0 \le |\gamma^{(\nu)}_{y_py_q}(m)|^2 \le 1, \qquad (33)$$

which directly translates into an inherent stepsize normalization of the corresponding update equation [8]. In particular,

$|\gamma^{(\nu)}_{y_1y_2}(m)|^2 = 0$ if $\mathbf{y}_1$ and $\mathbf{y}_2$ are orthogonal, and $|\gamma^{(\nu)}_{y_1y_2}(m)|^2 = 1$ when $\mathbf{y}_1 = a\mathbf{y}_2$ for any non-zero complex number a.

Comparing our update equation (31) with that derived in [8], we see that an additional approximation of $\left(\mathbf{S}^{(\nu)}_{yy}\right)^{-1}$ as a diagonal matrix was used in [8], which results in

$$\begin{aligned}\nabla_{\underline{\mathbf{W}}}\mathcal{J}^{(\nu)}(m) &= 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}^{(\nu)}_{xy}\operatorname{diag}^{-1}\mathbf{S}^{(\nu)}_{yy} \\ &\quad \cdot\left\{\mathbf{S}^{(\nu)}_{yy} - \operatorname{diag}\mathbf{S}^{(\nu)}_{yy}\right\}\operatorname{diag}^{-1}\mathbf{S}^{(\nu)}_{yy}. \quad (34)\end{aligned}$$

In the following, we study this link more closely using the so-called generalized coherence [12]

$$|\gamma^{(\nu)}_{yy}(m)|^2 = 1 - \frac{\det\mathbf{S}^{(\nu)}_{yy}(m)}{\prod_{p=1}^{P} S^{(\nu)}_{y_py_p}(m)}, \quad (35)$$

which is valid for an arbitrary number P of channels. By the Schwarz inequality, it can be shown that (35) also satisfies (33). In the 2×2 case it is equal to the well-known coherence (32). The generalized coherence can again be interpreted in a geometric way as in Fig. 2 since $\det\mathbf{S}^{(\nu)}(m)$ corresponds to a volume of a general parallelepiped spanned by the column vectors of $\mathbf{S}^{(\nu)}(m)$. Moreover, it is normalized by the volume $\prod_{p=1}^{P} S^{(\nu)}_{y_py_p}(m)$ of a (rectangular) P-dimensional cuboid.

To see the exact correspondence, we start with (14):

$$\begin{aligned}\mathcal{J}^{(\nu)}(m) &= \sum_{i=0}^{m}\beta(i,m)\Big\{\log\det\operatorname{diag}\mathbf{S}^{(\nu)}_{yy}(i) \\ &\quad -\log\det\mathbf{S}^{(\nu)}_{yy}(i)\Big\} \\ &= \sum_{i=0}^{m}\beta(i,m)\Big\{\log\prod_{p=1}^{P} S^{(\nu)}_{y_py_p}(i) \\ &\quad -\log\det\mathbf{S}^{(\nu)}_{yy}(i)\Big\} \\ &= \sum_{i=0}^{m}\beta(i,m)\left\{-\log\frac{\det\mathbf{S}^{(\nu)}_{yy}(i)}{\prod_{p=1}^{P} S^{(\nu)}_{y_py_p}(i)}\right\}. \quad (36)\end{aligned}$$

A Taylor approximation

$$-\log(x) = (1-x) + \frac{(1-x)^2}{2} + \frac{(1-x)^3}{3} + \cdots$$

around $x = 1$ for $0 < x \leq 2$ finally yields

$$\mathcal{J}^{(\nu)}(m) = \sum_{i=0}^{m}\beta(i,m)\left\{1 - \frac{\det\mathbf{S}^{(\nu)}_{yy}(m)}{\prod_{p=1}^{P} S^{(\nu)}_{y_py_p}(m)}\right\}.$$

For the case $P = 2$ this is exactly the cost function proposed in [8], while for $P > 2$ it is slightly more general. From this equivalence, we can draw the conclusion that our cost function (14) also leads to an inherent stepsize normalization for the coefficient updates.

4. CONCLUSIONS

We presented a generalization of a class of BSS algorithms for convolutive mixtures taking into account the nonstationarity property and the nonwhiteness property. Using a matrix framework and a generalized cost function, rigorous derivations of both known and new extended algorithms become possible.

5. REFERENCES

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley & Sons, Inc., New York, 2001.

[2] L. Molgedey and H. G. Schuster, “Separation of a mixture of independent signals using time delayed correlations,” *Physical Review Letters*, vol. 72, pp. 3634-3636, 1994.

[3] E. Weinstein, M. Feder, and A. Oppenheim, “Multi-channel signal separation by decorrelation,” *IEEE Trans. on Speech and Audio Processing*, vol 1, no. 4, pp. 405-413, Oct. 1993.

[4] S. Van Gerven and D. Van Compernolle,“Signal separation by symmetric adaptive decorrelation: stability, convergence, and uniqueness,” *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1602-1612, 1995.

[5] M. Kawamoto, K. Matsuoka, and N. Ohnishi, “A method of blind separation for convolved non-stationary signals,” *Neurocomputing*, vol. 22, pp. 157-171, 1998.

[6] L. Parra and C. Spence, “Convolutive blind source separation based on multiple decorrelation,” *Int. Workshop on Neural Networks for Signal Processing*, Cambridge, UK, Sept. 1998.

[7] H.-C. Wu and J. C. Principe,“Simultaneous diagonalization in the frequency domain (SDIF) for source separation,” in *Proc. ICA*, pp. 245-250, 1999.

[8] C.L. Fancourt and L. Parra, “The coherence function in blind source separation of convolutive mixtures of non-stationary signals,” in Proc. *Int. Workshop on Neural Networks for Signal Processing*, 2001.

[9] T. Nishikawa, H. Saruwatari, and K. Shikano, “Comparison of time-domain ICA, frequency-domain ICA and multistage ICA for blind source sepoaration,” in *Proc. European Signal Processing Conference*, vol. 2, pp. 15-18, Sep. 2002.

[10] R. Aichner et al., “Time-domain blind source separation of non-stationary convolved signals with utilization of geometric beamforming,” in Proc. *Int. Workshop on Neural Networks for Signal Processing*, Martigny, Switzerland, 2002.

[11] H. Buchner, J. Benesty, and W. Kellermann, “Multichannel Frequency-Domain Adaptive Algorithms with Application to Acoustic Echo Cancellation,” in J.Benesty and Y.Huang (eds.), *Adaptive signal processing: Application to real-world problems*, Springer-Verlag, Berlin/Heidelberg, Jan. 2003.

[12] H. Gish and D. Cochran, “Generalized Coherence,” *Proc. IEEE ICASSP*, pp. 2745-2748, 1988.

[13] A. Oppenheim, “Inequalities connected with definite hermitian forms,” *J. London Math. Soc.*, vol. 5, pp. 114-119, 1930.

[14] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, Springer-Verlag, Berlin, 1976.

[15] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, Wiley & Sons, Ltd., Chichester, UK, 2002.

[16] S. Amari, A. Cichocki, and H. H. Yang, “A new learning algorithm for blind signal separation,” in *Advances in neural information processing systems*, 8, Cambridge, MA, MIT Press, 1996, pp. 757-763.

[17] J.-F. Cardoso, “Blind signal separation: Statistical principles,” *Proc. IEEE*, vol. 86, pp. 2009-2025, Oct. 1998.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins, Baltimore, MD, 1989.