Accepted for Independent Component Analysis and Blind Signal Separation ICA 2003, April 4-6, 2003, Nara, Japan

FREQUENCY DOMAIN REALIZATION OF A MULTICHANNEL BLIND DECONVOLUTION ALGORITHM BASED ON THE NATURAL GRADIENT

Marcel Joho[1] *& Philip Schniter*[2]

[1]Phonak Inc., Champaign, IL, USA, (joho@ieee.org)
[2]Dept. of EE, The Ohio State University, Columbus, OH, USA (schniter.1@osu.edu)

ABSTRACT

This paper describes two efficient realizations of an adaptive multichannel blind deconvolution algorithm based on the natural gradient algorithm originally proposed by Amari, Douglas, Cichocki, and Yang. The proposed algorithms use fast convolution and correlation techniques and operate primarily in the frequency domain. Since the cost function minimized by the algorithms is well-defined in the time domain, the algorithms do not suffer from the so-called frequency-domain permutation problem. The proposed algorithm can be viewed as an multi-channel extension of a single-channel blind deconvolution algorithm recently proposed by the authors.

1. INTRODUCTION

1.1. Problem description and preliminary work

A well-known problem in acoustics is the so-called *cocktail party problem*, where the primary task is separation of a convolutive mixture of independently generated speech signals. A similar scenario occurs in data communications when an antenna array receives a multi-path mixture of signals transmitted from several sources. Independent of any application, this task is generally known as the *multichannel blind deconvolution* (MCBD) problem. The general setup is shown in Fig. 1. The sequence of the *source signals* $\mathbf{s} \triangleq \{\mathbf{s}_t\}$ is mixed and filtered by a causal *convolutive mixing system* $\mathbf{A} \triangleq \{\mathbf{A}_0, \mathbf{A}_1, \ldots\}$. This process can be described by a convolutional sum

$$\mathbf{x}_t = (\mathbf{A} * \mathbf{s})_t + \mathbf{n}_t = \sum\nolimits_{n=0}^{\infty} \mathbf{A}_n \mathbf{s}_{t-n} + \mathbf{n}_t \qquad (1)$$

where $\mathbf{x} \triangleq \{\mathbf{x}_t\}$ is the sequence of the *sensor signals* and $\mathbf{n} \triangleq \{\mathbf{n}_t\}$ is an *additive sensor noise* sequence. In a blind setup, only $\mathbf{x}$ is accessible to the algorithm; $\mathbf{s}$, $\mathbf{A}$, and $\mathbf{n}$ are unknown.

In the *multichannel blind deconvolution* problem we aim at finding a deconvolution filter $\mathbf{W} \triangleq \{\mathbf{W}_l\}$ such that the output $\mathbf{u} \triangleq \{u_t\}$ of the deconvolution process

$$\mathbf{u}_t = \sum_{l=-\infty}^{\infty} \mathbf{W}_l \mathbf{x}_{t-l} = \sum_{l=-\infty}^{\infty} (\mathbf{G}_l \mathbf{s}_{t-l} + \mathbf{W}_l \mathbf{n}_{t-l}) \qquad (2)$$

retrieves a waveform-preserving estimate of $\mathbf{s}$, possibly delayed and permuted. The *global-system* is defined as $\mathbf{G} \triangleq \{\mathbf{G}_l\} = \mathbf{W} * \mathbf{A}$ with $\mathbf{G}_l = (\mathbf{W} * \mathbf{A})_l$. Ideally, at least in the noise-free case, we wish to find a deconvolution matrix $\mathbf{W}$ such that the global system can be decomposed into $\mathbf{G}(z) = \mathbf{D}(z)\mathbf{P}$ where $\mathbf{G}(z) \triangleq \sum_l \mathbf{G}_l z^{-l}$ is the two-sided z-transform of $\mathbf{G}$, $\mathbf{P}$ is a permutation matrix, and $\mathbf{D}(z) = \mathrm{diag}(d'_1 z^{-\tau_1}, \ldots, d'_M z^{-\tau_M})$ is a diagonal matrix with monomial diagonal elements. Constraining $\mathbf{P}$ to be a permutation matrix guarantees perfect separation of the output signals (i.e., no *interchannel interference* (ICI)), and constraining $\mathbf{D}(z)$ to have non-zero monomial entries guarantees perfect deconvolution of each channel (i.e., no *intersymbol interference* (ISI)). The parameters $\mathbf{P}$, $\{d'_m\}$, and $\{\tau_m\}$ are indeterminacies of the MCBD problem [1].

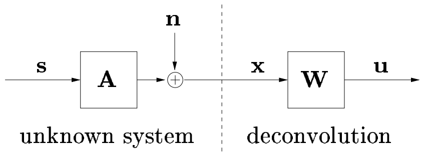

Figure 1: Setup of the multichannel blind deconvolution problem.

A convolutive mixing system can either be seen as the convolutive extension of instantaneous signal mixing or the multichannel extension of signal filtering, i.e., we can write $\mathbf{A}(z) = \sum_l \mathbf{A}_l$ or $\mathbf{A}(z) = [a_{mn}(z)]$. Hence, to solve the multichannel blind deconvolution problem, *blind source separation* (BSS) algorithms for instantaneous mixing systems and *single-channel blind deconvolution* (SCBD) algorithms must be properly merged. Alternatively, the BSS and SCBD problems can be regarded as being two special cases of the general MCBD problem, as shown in Fig. 2. For example, a combination of the BSS algorithm [2] and the SCBD algorithm [3] yielded the time-domain MCBD algorithm proposed by Amari *et al.* [4]. (See also the work of Douglas and Haykin [5] and [6].) Similar extensions in the frequency domain, carefully constructed so as not to suffer from the so-called *permutation problem*, have been carried out by Lambert [7, 8], Lambert

$$\begin{tikzcd}
\text{AGC} \arrow[r, "\text{spatial extension}"] \arrow[d, "\text{temporal extension}"'] & \text{BSS} \arrow[d, "\text{temporal extension}"] \\
\text{SCBD} \arrow[r, "\text{spatial extension}"'] & \text{MCBD}
\end{tikzcd}$$

Figure 2: Commutative diagram to reveal the relationship between different blind problems: Automatic gain control (AGC), blind source separation (BSS), single channel blind deconvolution (SCBD), and multichannel blind deconvolution (MCBD).

This work was funded in part by the Swiss Federal Institute of Technology, ETH Zurich, Switzerland.

and Bell [9], and Joho *et al.* [10]. We note that, in some applications of MCBD, perfect deconvolution is not required or sometimes not even desired. Algorithms for this latter case are presented in [11–14].

In the following, we present a low-complexity implementation of the time-domain MCBD algorithm proposed by Amari *et al.* in [4]. The underlying cost function is defined in the time domain and assumes that that the source signals $(\mathbf{s})_m \triangleq \{s_{m,t}\}$ are mutually independent, white, and non-Gaussian. For the derivation of the algorithm, we will restrict ourselves to the case that $\mathbf{A}$ and $\mathbf{W}$ are both $M \times M$ dimensional systems and $\mathbf{W}$ is causal finite impulse response (FIR). We allow the source signals $\{s_{m,t}\}$ and the mixing system $\mathbf{A}$ to be complex valued. For computational efficiency reasons, the convolution operations are implemented in the frequency domain. However, the cost function, and therefore also the non-linearity, remains in the time domain to avoid a bin-wise permutation indeterminacy. Consequently, the proposed algorithm can be viewed as an extension of the SCBD algorithm recently proposed by the authors in [15].

The outline of the paper is as follows: In Section 2 we describe the time domain algorithm, in Section 3 we derive the proposed frequency domain algorithm, in Section 4 we give a simulation example, and in Section 5 we draw conclusions. In Appendix A we summarize some useful properties of circulant matrices.

1.2. Notation

The notation used throughout this paper is the following: Vectors are written in lower case bold, matrices in upper case bold. Matrix and vector transpose, complex conjugation and Hermitian transpose are denoted by $(.)^T$, $(.)^*$, and $(.)^H = ((.)^*)^T$, respectively. The element-wise multiplication of two vectors or matrices is denoted by $\odot$. The identity matrix is denoted by $\mathbf{I}$ and a vector or matrix containing only zeros is denoted by $\mathbf{0}$. The C-point DFT matrix $\mathbf{F}_C$ is defined as

$$[\mathbf{F}_C]_{mn} \triangleq e^{-j\frac{2\pi}{C}mn} \qquad (m,n = 0\dots C-1) \tag{3}$$

$$\mathbf{F}_C^{-1} = 1/C \cdot \mathbf{F}_C^H = 1/C \cdot \mathbf{F}_C^*. \tag{4}$$

Circulant and diagonal matrices are denoted as $\tilde{\mathbf{A}}$ and $\bar{\mathbf{A}}$, respectively, and $\bar{\mathbf{a}} \triangleq \mathrm{diag}(\bar{\mathbf{A}})$ denotes a vector containing the diagonal elements of $\bar{\mathbf{A}}$. The operation $\tilde{\mathbf{A}} \triangleq \mathcal{C}(\tilde{\mathbf{a}})$ defines a circulant matrix with $\tilde{\mathbf{a}}$ in its first column and $\tilde{\mathbf{a}} \triangleq \mathcal{C}^{-1}(\tilde{\mathbf{A}})$ is the corresponding inverse operation (see Appendix A for more details).

The variable t is used as a discrete sample index. The block index is denoted by $[k] \triangleq (k \cdot L)$ where L is the block length (block forward shift).

2. TIME-DOMAIN IMPLEMENTATION

2.1. Instantaneous mixing

In [2] Amari *et al.* proposed the following BSS algorithm for the instantaneous mixing case

$$\mathbf{u}_t = \mathbf{W}_t \mathbf{x}_t \tag{5}$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mu(\mathbf{I} - \mathbf{y}_t \mathbf{u}_t^H)\mathbf{W}_t \tag{6}$$

where $y_{m,t} = g_m(u_{m,t})$. This algorithm is based on the *natural-gradient* learning algorithm. Based on a local convergence analysis, the optimal nonlinearity $g_m(.)$ is suggested to be the *score function* of the pdf of the source s_m [16]. However, simulations have shown that the performance of the algorithm is relatively robust to selection of $g_m(.)$. Typical choices are

$$g_m(u_{m,t}) \propto \mathrm{sign}(u_{m,t}) \tag{7}$$

for *super-Gaussian* signals (kurtosis larger than for a Gaussian signal) and

$$g_m(u_{m,t}) \propto u_{m,t}|u_{m,t}|^2 \tag{8}$$

for *sub-Gaussian* signals (kurtosis smaller than for a Gaussian signal). For convenience, we require that $g_m(0) \equiv 0$.

2.2. Convolutive mixing

The BSS algorithm (5)–(6) was extended by Amari *et al.* in [4] to the case of convolutive mixing. Their *time-domain multichannel blind deconvolution* algorithm (TDMCBD) is defined, for time index t, as

$$\mathbf{u}_t = \sum\nolimits_{l=0}^{N} \mathbf{W}_l(t)\, \mathbf{x}_{t-l} \tag{9}$$

$$\mathbf{v}_t = \sum\nolimits_{l=0}^{N} \mathbf{W}_{N-l}^H(t)\, \mathbf{u}_{t-l} \tag{10}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{u}_t) \tag{11}$$

$$\mathbf{W}_l(t+1) = \mathbf{W}_l(t) + \mu\, \left(\mathbf{W}_l(t) - \mathbf{y}_{t-N}\mathbf{v}_{t-l}^H\right),\ l \in [0,N] \tag{12}$$

where $\{\mathbf{u}_t\}$ contains the output samples and $\mathbf{v}_t$ is an intermediate signal. The *multichannel deconvolution filter* $\{\mathbf{W}_l\}$ is an multichannel FIR filter of length $N+1$. Note that, for $N=0$, the TDMCBD algorithm (9)–(12) becomes identical to the BSS algorithm (5)–(6). On the other hand, for $M=1$, (9)–(12) becomes the single-channel blind deconvolution algorithm proposed in [3].

2.3. Block-wise filtering and adaptation

Alternatively, we can carry out the filtering and the adaptation of the TDMCBD algorithm in a block-wise manner. At block k, i.e., $t = kL - L + 1, \dots, kL$, we have

$$\mathbf{u}_t = \sum\nolimits_{l=0}^{N} \mathbf{W}_l[k]\, \mathbf{x}_{t-l} \tag{13}$$

$$\mathbf{v}_t = \sum\nolimits_{l=0}^{N} \mathbf{W}_{N-l}^H[k]\, \mathbf{u}_{t-l} \tag{14}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{u}_t) \tag{15}$$

$$\mathbf{W}_l[k+1] = (1+\mu)\,\mathbf{W}_l[k] - \frac{\mu}{L} \cdot \sum_{t=kL-L+1}^{kL} \mathbf{y}_{t-N}\mathbf{v}_{t-l}^H \tag{16}$$

where L is the block size (block-wise forward shift). The block-wise update in (16) is equal to the average of the sample-wise update in (12) over the entire block of L samples. We will refer to the algorithm (13)–(16) as the *block time-domain multichannel blind deconvolution* algorithm (BTDMCBD).

In the following, we will restrict ourselves to the case where $N = L$, which will simplify the derivation. We rewrite (13), (14), and (16) for block k in matrix form:

$$\begin{bmatrix} u_{m,kL-3L+1} \\ \vdots \\ u_{m,kL} \end{bmatrix} = \sum_{n=1}^{M} \begin{bmatrix} w_{mnL} \cdots w_{mn0} & & \\ & \ddots \quad \ddots \quad \ddots & \\ & & w_{mnL} \cdots w_{mn0} \end{bmatrix} \cdot \begin{bmatrix} x_{n,kL-4L+1} \\ \vdots \\ x_{n,kL} \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} v_{m,kL-2L+1} \\ \vdots \\ v_{m,kL} \end{bmatrix} = \sum_{n=1}^{M} \begin{bmatrix} w_{nm0} \cdots w_{nmL} & & \\ & \ddots \quad \ddots \quad \ddots & \\ & & w_{nm0} \cdots w_{nmL} \end{bmatrix}^{*} \cdot \begin{bmatrix} u_{n,kL-3L+1} \\ \vdots \\ u_{n,kL} \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} w_{mn0}[k+1] \\ \vdots \\ w_{mnL}[k+1] \end{bmatrix} = (1+\mu) \begin{bmatrix} w_{mn0}[k] \\ \vdots \\ w_{mnL}[k] \end{bmatrix} - \frac{\mu}{L} \begin{bmatrix} v_{n,kL-L+1} & \cdots & v_{n,kL} \\ v_{n,kL-L} & \ddots & \vdots \\ \vdots & \ddots & v_{n,kL-L+1} \\ v_{n,kL-2L+1} & \cdots & v_{n,kL-L} \end{bmatrix}^{*} \cdot \begin{bmatrix} y_{m,kL-2L+1} \\ \vdots \\ y_{m,kL-L} \end{bmatrix} . \tag{19}$$

In (17) and (18) we have omitted the block index $[k]$ for the filter coefficients and, hence, w_{mnl} stands for $w_{mnl}[k]$. Furthermore

$$\mathbf{w}_{mn}[k] \triangleq (w_{mn0}[k], \ldots, w_{mnL}[k])^T \tag{20}$$

is the impulse response of the mn-th deconvolution filter at block k.

In block k the update equation (16) needs to be evaluated for $t = kL - L + 1, \ldots, kL$. This determines the dimensionality of (19). In (19) we need the latest $2L$ samples of v_n, i.e., $\{v_{n,kL-2L+1}, \ldots, v_{n,kL}\}$, which then determines the dimensionality of the LHS vector in (18). For the same reason, the LHS vector in (17) needs to be the same as the RHS vector in (18). Eq. (17) and (18) guarantee that all signal samples used for the update in (19) are derived from the current $\mathbf{W}[k] = \{\mathbf{W}_l[k]\} = \{[\mathbf{w}_{mnl}[k]]\}$. The equations (17)–(19), together with $y_{m,t} = g_m(u_{m,t})$, yield the BTDMCBD algorithm in matrix form.

3. FREQUENCY-DOMAIN IMPLEMENTATION

3.1. FDMCBD-I: 75% overlap

In the following, we employ fast convolution techniques to reduce the computationally complexity of the BTDMCBD algorithm. To this end, we define the following vectors of length L:

$$\mathbf{x}_{m,k} \triangleq (x_{m,kL-L+1}, \ldots, x_{m,kL})^T \tag{21}$$

$$\mathbf{u}_{m,k} \triangleq (u_{m,kL-L+1}, \ldots, u_{m,kL})^T \tag{22}$$

$$\mathbf{v}_{m,k} \triangleq (v_{m,kL-L+1}, \ldots, v_{m,kL})^T \tag{23}$$

$$\mathbf{y}_{m,k} \triangleq g_m(\mathbf{u}_{m,k}) . \tag{24}$$

where the nonlinearity function $g_m(.)$ is applied on each vector element. We also define the following vectors of length $4L$:

$$\tilde{\mathbf{w}}_{mn}[k] \triangleq (\mathbf{w}_{mn}^T[k], \mathbf{0}_{3L-1}^T)^T \tag{25}$$

$$\tilde{\mathbf{x}}_{m,k} \triangleq (\mathbf{x}_{m,k-3}^T, \mathbf{x}_{m,k-2}^T, \mathbf{x}_{m,k-1}^T, \mathbf{x}_{m,k}^T)^T \tag{26}$$

$$\tilde{\mathbf{u}}_{m,k} \triangleq (\dot{\mathbf{u}}_{m,k-3}^T, \mathbf{u}_{m,k-2}^T, \mathbf{u}_{m,k-1}^T, \mathbf{u}_{m,k}^T)^T \tag{27}$$

$$\tilde{\mathbf{v}}_{m,k} \triangleq (\dot{\mathbf{v}}_{m,k-2}^T, \mathbf{v}_{m,k-1}^T, \mathbf{v}_{m,k}^T, \dot{\mathbf{v}}_{m,k-4}^T)^T \tag{28}$$

$$\tilde{\mathbf{y}}_{m,k} \triangleq (\mathbf{0}_L^T, \mathbf{0}_L^T, \mathbf{y}_{m,k-1}^T, \mathbf{0}_L^T)^T . \tag{29}$$

We denote a vector that can contain arbitrary elements with a dot e.g., $\dot{\mathbf{u}}_{k-3}$. Furthermore, we define the following circulant matrices

$$\tilde{\mathbf{W}}_{mn}[k] \triangleq \mathcal{C}(\tilde{\mathbf{w}}_{mn}[k]) \tag{30}$$

$$\tilde{\mathbf{U}}_{m,k} \triangleq \mathcal{C}(\tilde{\mathbf{u}}_{m,k}) \tag{31}$$

$$\tilde{\mathbf{V}}_{m,k} \triangleq \mathcal{C}(\tilde{\mathbf{v}}_{m,k}) \tag{32}$$

and the projection matrices

$$\mathbf{P}_{\mathbf{w}} \triangleq \begin{bmatrix} \mathbf{I}_{L+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{3L-1} \end{bmatrix} \tag{33}$$

$$\mathbf{P}_{\mathbf{y}} \triangleq \begin{bmatrix} \mathbf{0}_{2L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_L \end{bmatrix} . \tag{34}$$

Since all the matrices involved in the equations (17) to (19) are Toeplitz, we can enlarge them to $4L \times 4L$ circulant matrices, such that we can embed (17)–(19) for block k in the following equations

$$\tilde{\mathbf{u}}_{m,k} = \sum\nolimits_{n=1}^{M} \tilde{\mathbf{W}}_{mn}[k] \, \tilde{\mathbf{x}}_{n,k} \tag{35}$$

$$\tilde{\mathbf{v}}_{m,k} = \sum\nolimits_{n=1}^{M} \tilde{\mathbf{W}}_{nm}^H \, \tilde{\mathbf{u}}_{n,k} \tag{36}$$

$$\tilde{\mathbf{y}}_{m,k} = g_m(\mathbf{P}_{\mathbf{y}} \, \tilde{\mathbf{u}}_{m,k}) \tag{37}$$

$$\tilde{\mathbf{w}}_{mn}[k+1] = (1+\mu) \, \tilde{\mathbf{w}}_{mn}[k] - \frac{\mu}{L} \mathbf{P}_{\mathbf{w}} \tilde{\mathbf{V}}_{n,k}^H \, \tilde{\mathbf{y}}_{m,k} . \tag{38}$$

Recall that we required $g_m(0) \equiv 0$. Therefore $\tilde{\mathbf{y}}_{m,k}$ will have the zero padded structure as given in (29). Since (35), (36), and (38) describe now circular convolutions, fast convolution techniques can now be employed. Towards this end, we define the following vectors of length $C = 4L$:

$$\bar{\mathbf{w}}_{mn}[k] \triangleq \mathbf{F}_C \, \tilde{\mathbf{w}}_{mn}[k] = \mathrm{FFT}(\tilde{\mathbf{w}}_{mn}[k]) \tag{39}$$

$$\bar{\mathbf{x}}_{m,k} \triangleq \mathbf{F}_C \, \tilde{\mathbf{x}}_{m,k} = \mathrm{FFT}(\tilde{\mathbf{x}}_{m,k}) \tag{40}$$

$$\bar{\mathbf{u}}_{m,k} \triangleq \mathbf{F}_C \, \tilde{\mathbf{u}}_{m,k} = \mathrm{FFT}(\tilde{\mathbf{u}}_{m,k}) \tag{41}$$

$$\bar{\mathbf{v}}_{m,k} \triangleq \mathbf{F}_C \, \tilde{\mathbf{v}}_{m,k} = \mathrm{FFT}(\tilde{\mathbf{v}}_{m,k}) \tag{42}$$

$$\bar{\mathbf{y}}_{m,k} \triangleq \mathbf{F}_C \, \tilde{\mathbf{y}}_{m,k} = \mathrm{FFT}(\tilde{\mathbf{y}}_{m,k}) . \tag{43}$$

Consequently we have from (39)

$$\tilde{\mathbf{w}}_{mn}[k] \triangleq \mathbf{F}^{-1} \bar{\mathbf{w}}_{mn}[k] = \mathrm{IFFT}(\bar{\mathbf{w}}_{mn}[k]) . \tag{44}$$

The same is also true for inverting (40)–(43).

We now wish to transform (35)–(38) into the frequency domain. To this end, we pre-multiply the equations on both side with the Fourier matrix $\mathbf{F}$. We begin with (35)

$$\bar{\mathbf{u}}_{m,k} = \sum\nolimits_{n=1}^{M} \mathbf{F} \, \tilde{\mathbf{W}}_{mn}[k] \, \mathbf{F}^{-1} \, \mathbf{F} \, \tilde{\mathbf{x}}_{n,k} \tag{45}$$

$$= \sum\nolimits_{n=1}^{M} \bar{\mathbf{W}}_{mn}[k] \, \bar{\mathbf{x}}_{n,k} \tag{46}$$

where $\bar{\mathbf{W}}_{mn}[k] \triangleq \mathbf{F} \, \tilde{\mathbf{W}}_{mn}[k] \, \mathbf{F}^{-1}$. Note that from (61) we know that $\bar{\mathbf{W}}_{mn}[k] = \mathrm{diag}(\mathbf{F} \, \tilde{\mathbf{w}}_{mn}[k])$ is a diagonal matrix. Applying similar steps to (36) and using (62) we get

$$\bar{\mathbf{v}}_k = \sum\nolimits_{n=1}^{M} \mathbf{F} \, \tilde{\mathbf{W}}_{nm}^H[k] \, \mathbf{F}^{-1} \, \mathbf{F} \, \tilde{\mathbf{u}}_{n,k} \tag{47}$$

$$= \sum\nolimits_{n=1}^{M} \bar{\mathbf{W}}_{nm}^{*}[k] \, \bar{\mathbf{u}}_{n,k} . \tag{48}$$

Transforming (37) into the frequency domain gives

$$\bar{\mathbf{y}}_{m,k} = \mathbf{F}\, g_m\left(\mathbf{P}_{\mathbf{y}}\,\mathbf{F}^{-1}\,\bar{\mathbf{u}}_{m,k}\right) \tag{49}$$

and applying property (62) to (38) gives

$$\begin{aligned}\bar{\mathbf{w}}_{mn}[k+1] &= (1+\mu)\,\bar{\mathbf{w}}_{mn}[k] \\ &\quad - \frac{\mu}{L}\,\mathbf{F}\,\mathbf{P}_{\mathbf{w}}\,\mathbf{F}^{-1}\,\mathbf{F}\,\tilde{\mathbf{V}}^{H}_{n,k}\,\mathbf{F}^{-1}\,\mathbf{F}\,\tilde{\mathbf{y}}_{m,k} & (50)\\ &= (1+\mu)\,\bar{\mathbf{w}}_{mn}[k] - \frac{\mu}{L}\,\mathbf{F}\,\mathbf{P}_{\mathbf{w}}\,\mathbf{F}^{-1}\,\bar{\mathbf{V}}^{*}_{n,k}\,\bar{\mathbf{y}}_{m,k}\,. & (51)\end{aligned}$$

Equations (46), (48), and (51) can be rewritten as

$$\bar{\mathbf{u}}_{m,k} = \sum\nolimits_{n=1}^{M} \bar{\mathbf{w}}_{mn}[k] \odot \bar{\mathbf{x}}_{n,k} \tag{52}$$

$$\bar{\mathbf{v}}_{m,k} = \sum\nolimits_{n=1}^{M} \bar{\mathbf{w}}^{*}_{nm}[k] \odot \bar{\mathbf{u}}_{n,k} \tag{53}$$

$$\bar{\mathbf{w}}_{mn}[k+1] = (1+\mu)\,\bar{\mathbf{w}}_{mn}[k] - \frac{\mu}{L}\mathbf{F}\,\mathbf{P}_{\mathbf{w}}\mathbf{F}^{-1}\left(\bar{\mathbf{v}}^{*}_{n,k} \odot \bar{\mathbf{y}}_{m,k}\right). \tag{54}$$

Alternatively, since $\mathbf{P}_{\mathbf{w}}\tilde{\mathbf{w}}_{mn}[k] = \tilde{\mathbf{w}}_{mn}[k]$ holds, we can reformulate (54) as

$$\bar{\mathbf{w}}_{mn}[k+1] = \mathbf{F}\,\mathbf{P}_{\mathbf{w}}\,\mathbf{F}^{-1}\left((1+\mu)\,\bar{\mathbf{w}}_{mn}[k] - \frac{\mu}{L}\,\bar{\mathbf{v}}^{*}_{n,k} \odot \bar{\mathbf{y}}_{m,k}\right). \tag{55}$$

We will refer to (52), (53), (49), and (55) as the FDMCBD-I algorithm The complete implementation of the filter and adaptation equations is summarized in Fig. 3. We have preferred to use (55) over (54), which has the advantage that wrap-around errors do not accumulate in $\tilde{\mathbf{w}}_{mn}[k]$ if the projection operation $\mathbf{F}\,\mathbf{P}_{\mathbf{w}}\,\mathbf{F}^{-1}$ is not carried out in each block (e.g., alternated filter projections [17]).

In the derivation of the proposed algorithm we have restricted ourselves to the case with $N = L$ and $C = 4L$. However, different choices for N are possible as long as $C \geq L + 3(N-1)$.

3.2. FDMCBD-I: 25% overlap

The FDMCBD-I algorithm as proposed in Sec. 3.1 has a 75% overlap between the samples of two subsequent input vectors $\tilde{\mathbf{x}}_{m,k-1}$ and $\tilde{\mathbf{x}}_{m,k}$. Therefore, an output block $\mathbf{u}_{m,k}$ is computed, in fact, three times, namely in block k, $k+1$, and $k+2$. On the other hand, the vector $\mathbf{y}_{m,k-1}$, which is used for updating the filter coefficients, is computed only once, namely in block k. It is also possible to define a variant of FDMCBD-I with only 25% overlap between subsequent input vectors by replacing (26) with

$$\tilde{\mathbf{x}}_{m,k} \triangleq \left(\mathbf{x}^T_{m,3k-3}, \mathbf{x}^T_{m,3k-2}, \mathbf{x}^T_{m,3k-1}, \mathbf{x}^T_{m,3k}\right)^T \tag{56}$$

and replacing (27) with

$$\tilde{\mathbf{u}}_{m,k} \triangleq \left(\dot{\mathbf{u}}^T_{m,3k-3}, \mathbf{u}^T_{m,3k-2}, \mathbf{u}^T_{m,3k-1}, \mathbf{u}^T_{m,3k}\right)^T. \tag{57}$$

The $3L$ output samples $(\mathbf{u}^T_{m,3k-2}, \mathbf{u}^T_{m,3k-1}, \mathbf{u}^T_{m,3k})^T$ from $\tilde{\mathbf{u}}_{m,k}$ are now computed only once. Essentially, this is equivalent to computing the filtering and adaptation steps only at blocks $\{\ldots, k-3, k, k+3, \ldots\}$. This simple trick reduces the computational complexity by a factor of three. (See also [15].)

FDMCBD-I

Definitions:

$$\mathbf{P}_{\mathbf{w}} \triangleq \begin{bmatrix} \mathbf{I}_{L+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{3L-1} \end{bmatrix}$$

$$\mathbf{P}_{\mathbf{y}} \triangleq \begin{bmatrix} \mathbf{0}_{2L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{L} \end{bmatrix}$$

Initialization ($\forall m, n$):

$$\begin{aligned}\mathbf{w}_{mn}[0] &\triangleq (w_{mn0}[0], \ldots, w_{mnL}[0])^T \\ \tilde{\mathbf{w}}_{mn}[0] &\triangleq (\mathbf{w}^T_{mn}[0], \mathbf{0}^T_{3L-1})^T \\ \bar{\mathbf{w}}_{mn}[0] &\triangleq \mathrm{FFT}(\tilde{\mathbf{w}}_{mn}[0])\end{aligned}$$

For each loop k do:

Filtering ($\forall m$):

$$\begin{aligned}\mathbf{x}_{m,k} &\triangleq (x_{m,kL-L+1}, \ldots, x_{m,kL})^T \\ \tilde{\mathbf{x}}_{m,k} &\triangleq (\mathbf{x}^T_{m,k-3}, \mathbf{x}^T_{m,k-2}, \mathbf{x}^T_{m,k-1}, \mathbf{x}^T_{m,k})^T \\ \bar{\mathbf{x}}_{m,k} &= \mathrm{FFT}(\tilde{\mathbf{x}}_{m,k}) \\ \bar{\mathbf{u}}_{m,k} &= \sum_{n=1}^{M} \bar{\mathbf{w}}_{mn}[k] \odot \bar{\mathbf{x}}_{n,k} \\ \tilde{\mathbf{u}}_{m,k} &\triangleq (\dot{\mathbf{u}}^T_{m,k-3}, \mathbf{u}^T_{m,k-2}, \mathbf{u}^T_{m,k-1}, \mathbf{u}^T_{m,k})^T \\ &= \mathrm{IFFT}(\bar{\mathbf{u}}_{m,k})\end{aligned}$$

Adaptation ($\forall m, n$):

$$\begin{aligned}\bar{\mathbf{v}}_{m,k} &= \sum_{n=1}^{M} \bar{\mathbf{w}}^{*}_{nm}[k] \odot \bar{\mathbf{u}}_{n,k} \\ \tilde{\mathbf{y}}_{m,k} &\triangleq (\mathbf{0}^T_L, \mathbf{0}^T_L, \mathbf{y}^T_{m,k-1}, \mathbf{0}^T_L)^T = g_m(\mathbf{P}_{\mathbf{y}}\,\tilde{\mathbf{u}}_{m,k}) \\ \bar{\mathbf{y}}_{m,k} &= \mathrm{FFT}(\tilde{\mathbf{y}}_{m,k}) \\ \bar{\mathbf{w}}'_{mn}[k+1] &= (1+\mu)\,\bar{\mathbf{w}}_{mn}[k] - \frac{\mu}{L}\,\bar{\mathbf{y}}_{m,k} \odot \bar{\mathbf{v}}^{*}_{n,k} \\ \bar{\mathbf{w}}_{mn}[k+1] &= \mathrm{FFT}\left(\mathbf{P}_{\mathbf{w}}\ \mathrm{IFFT}\left(\bar{\mathbf{w}}'_{mn}[k+1]\right)\right)\end{aligned}$$

Figure 3: FDMCBD-I 75% Algorithm (FFT size $C = 4L$).

4. SIMULATION EXAMPLE

In the following, we give a simulation example to examine the behavior of the FDMCBD-I 75% algorithm. A real-valued 2 × 2 convolutive mixing system $\mathbf{A}(z)$ is chosen randomly. Fig. 4 shows some properties of $\mathbf{A}(z)$, e.g., the locations of the poles of $1/\det(\mathbf{A}(z))$. They indicate how difficult the mixing system is to invert, as $\mathbf{A}^{-1}(z) = \mathrm{adj}(\mathbf{A}(z))/\det(\mathbf{A}(z))$. It is clearly seen that $\mathbf{A}(z)$ is non-minimum phase, and, hence, $\mathbf{W}(z)$ will have to adapt to a two-sided filter. The transfer function $|1/\det(\mathbf{A}(e^{j\omega})|$ and the corresponding stable impulse response are also given in Fig. 4.

We carry out two simulations. In the first simulation, the source signals s_m are random 2-PAM (sub-Gaussian) signals,

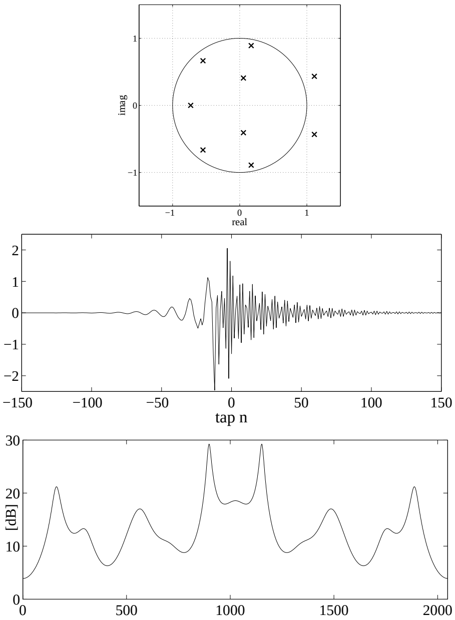

Figure 4: Roots of $\det(A(z))$ (top), two-sided impulse response of $1/\det(A(z))$ (middle), and transfer function of $|1/\det(A(e^{j\omega}))|$.

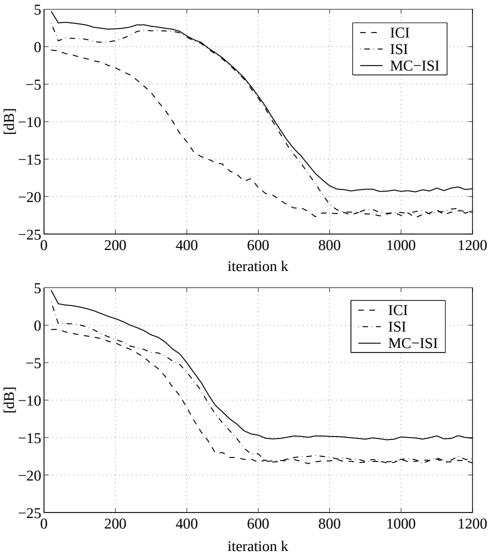

Figure 5: Convergence behavior of the FDMCBD-I 75% algorithm for sub-Gaussian (top) and super-Gaussian (bottom) source signals. The performance measures are output ICI, ISI, and MC-ISI.

and the nonlinearity is $g_m(u_{m,t}) = u_{m,t}\,|u_{m,t}|^2$. In the second simulation, the source signals s_m are random two-sided gamma-distributed (super-Gaussian) signals, and the nonlinearity is $g_m(u_{m,t}) = \text{sign}(u_{m,t})$. The block size is $L = 512$, the FFT size is $C = 4L = 2048$, and, hence, $4(L+1) = 2052$ filter coefficients are adapted. The diagonal elements of $\mathbf{W}(z)$ are initialized with a center spike, the off-diagonal element filters are set to zero. For both simulations the step size is $\mu = 0.03$ and the input SNR is 30 dB.

The performance curves for a single run are shown in Fig. 5. The performance criteria are the average residual *interchannel interference* (ICI) and *intersymbol interference* (ISI), as defined in [10], and also the multichannel-ISI (MC-ISI) [18]. As seen in both cases, the ICI performance reveals a slightly faster convergence behavior than the ISI. However, the residual values eventually converge to the same value. Thus, the MC-ISI yields roughly a 3 dB higher value, as it is approximately the sum of the ICI and ISI after convergence [10].

The impulse responses and the corresponding transfer functions of the mixing matrix $\mathbf{A}(z)$, the demixing matrix $\mathbf{W}(z)$ after convergence, and the global system $\mathbf{G}(z) = \mathbf{W}(z)\mathbf{A}(z)$ are shown in Fig. 6. The shape of $\mathbf{G}(z)$ clearly shows the successful separation and deconvolution of the source signals. The reason why the transfer functions of $\mathbf{G}(z)$ show a slight ripple, is to compromise the strong noise amplification.

5. CONCLUSIONS

We have presented a low-complexity exact frequency-domain implementation of the time-domain multichannel blind deconvolution algorithm proposed by Amari *et al.* in [4]. The computational complexity of the algorithm can be reduced by a factor of three (at the expense of a factor-of-three decrease in convergence rate) simply by changing the overlap between consecutive input vectors from 75% to 25%. By suitable choice of nonlinearity, the algorithm can handle the case where all source signals are sub-Gaussian or the case where all signals are super-Gaussian. Even though most of the computation is carried out in the frequency domain, the algorithm does not suffer from a bin-wise permutation indeterminacy because the underlying cost function has been well-defined in the time domain. The presented algorithm is the multichannel extension of the single-channel algorithm in [15]. Simulation examples demonstrate the separation and deconvolution capabilities of the presented algorithm. Possible applications of the algorithm are in acoustics, e.g., teleconferencing or hearing aids.

6. REFERENCES

[1] L. Tong, R.-W. Liu, V. C. Soon, and Y.-F. Huang, “Indeterminacy and identifiability of blind identification,” *IEEE Trans. Circuits Syst.*, vol. 38, no. 5, pp. 499–509, May 1991.

[2] S.-I. Amari, A. Cichocki, and H. H. Yang, “A new learning algorithm for blind signal separation,” *Advances in Neural Information Processing Systems*, vol. 8, pp. 757–763, 1996.

[3] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, “Novel online adaptive learning algorithms for blind deconvolution using the natural gradient approach,” in *Proc. SYSID*, Kitakyushu, Japan, July 8–11, 1997, pp. 1057–1062.

[4] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, “Multichannel blind deconvolution and equalization using the natural gradient,” in *Proc. SPAWC*, Paris, France, Apr. 16–18, 1997, pp. 101–104.

[5] S. C. Douglas and S. Haykin, “Relationships between blind deconvolution and blind source separation,” in *Unsupervised Adaptive Filtering, Volume II: Blind Deconvolution*, S. Haykin, Ed. 2000, pp. 113–145, John Wiley & Sons.

[6] S. C. Douglas and S. Haykin, “On the relationship between blind deconvolution and blind source separation,” in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2–5, 1997, vol. II, pp. 1591–1595.

[7] R. H. Lambert, *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*, Ph.D. thesis, University of Southern California, 1996.

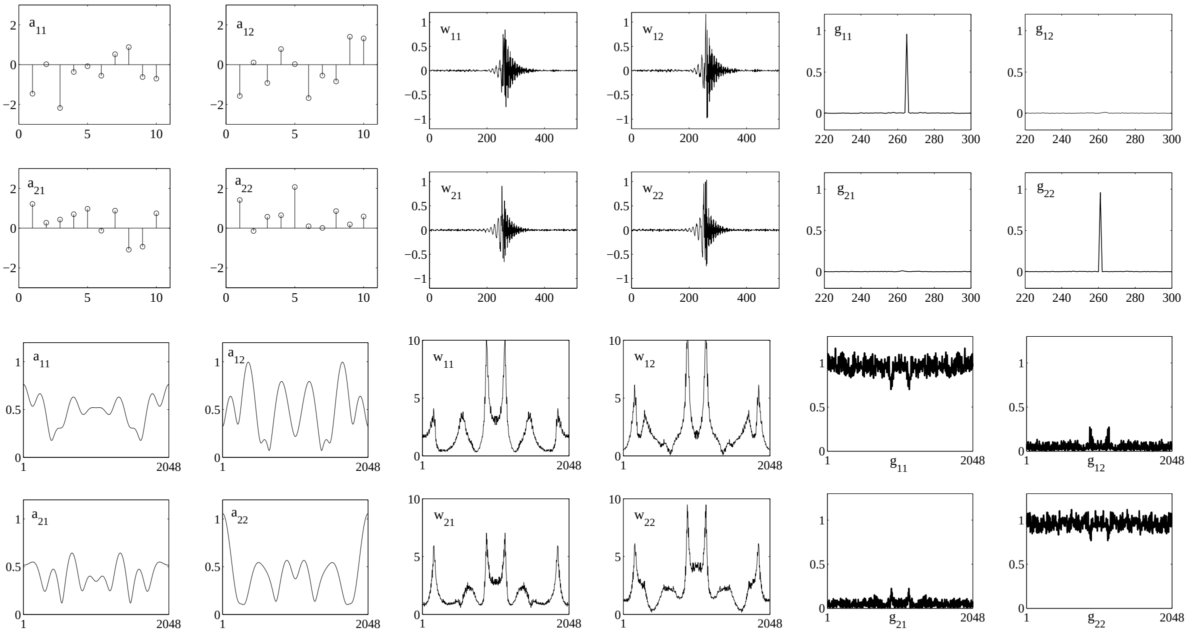

Figure 6: Impulse responses (top rows) and transfer functions (bottom rows) of $A(z)$ (left), $W(z)$ (middle), and $G(z)$ (right).

[8] R. H. Lambert and C. L. Nikias, "Blind deconvolution of multipath mixtures," in *Unsupervised Adaptive Filtering, Volume I: Blind Source Separation*, S. Haykin, Ed. 2000, pp. 377–436, John Wiley & Sons.

[9] R. H. Lambert and A. J. Bell, "Blind separation of multiple speakers in a multipath environment," in *Proc. ICASSP*, Munich, Germany, Apr. 21–24, 1997, pp. 423–426.

[10] M. Joho, H. Mathis, and G. S. Moschytz, "An FFT-based algorithm for multichannel blind deconvolution," in *Proc. ISCAS*, Orlando, FL, May 30 – June 2, 1999, pp. III–203–206.

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

[12] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, 2000.

[13] K. Rahbar and J. P. Reilly, "Blind source separation algorithm for MIMO convolutive mixtures," in *Proc. ICA*, San Diego, CA, Dec. 9–12, 2001, pp. 224–229.

[14] K. Rahbar, J. P. Reilly, and J. H. Manton, "A frequency domain approach to blind identification of mimo fir systems driven by quasi-stationary signals," in *Proc. ICASSP*, Orlando, FL, May 13–17, 2002, vol. 2, pp. 1717–1720.

[15] M. Joho and P. Schniter, "On frequency-domain implementations of filtered-gradient blind deconvolution algorithms," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 3–6, 2002.

[16] S. Amari and J. F. Cardoso, "Blind source separation—semiparametric statistical approach," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2692–2700, 1997.

[17] M. Joho and G. S. Moschytz, "Connecting partitioned frequency-domain filters in parallel or in cascade," *IEEE Trans. Circuits Syst.–II*, vol. 47, no. 8, pp. 685–698, Aug. 2000.

[18] R. H. Lambert, "Difficulty measures and figures of merit for source separation," in *Proc. ICA*, Aussois, France, Jan. 11–15, 1999, pp. 133–138.

[19] P. J. Davis, *Circulant Matrices*, John Wiley & Sons, 1979.

[20] R. M. Gray, *Toeplitz and Circulant Matrices: A review*, Stanford Electron. Lab., Tech. Rep. 6502-1, June 1971.

[21] M. Joho, *A Systematic Approach to Adaptive Algorithms for Multichannel System Identification, Inverse Modeling, and Blind Identification*, Ph.D. thesis, ETH Zürich, Dec. 2000.

7. ACKNOWLEDGMENT

The authors would like to thank Scott C. Douglas, Russell H. Lambert, Nail Çadallı, and Michael Kramer for helpful discussions.

A. CIRCULANT MATRICES AND BASIC PROPERTIES

Since there is a very close relationship between circulant matrix products and circular convolutions, we recall some of the basic properties of circulant matrices. For a thorough description we refer to [19, 20] or [21, Chapter 3].

Let $\tilde{\mathbf{a}} = (a_1, \ldots, a_C)^T$. We define the corresponding circulant matrix $\tilde{\mathbf{A}}$ which has $\tilde{\mathbf{a}}$ as its first column as

$$\tilde{\mathbf{A}} \triangleq \mathcal{C}(\tilde{\mathbf{a}}) \triangleq \begin{bmatrix} a_1 & a_C & \ldots & a_2 \\ a_2 & a_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_C \\ a_C & \ldots & a_2 & a_1 \end{bmatrix}. \tag{58}$$

The inverse operation $\tilde{\mathbf{a}} \triangleq \mathcal{C}^{-1}(\tilde{\mathbf{A}})$ returns the first column of $\tilde{\mathbf{A}}$. Furthermore, we define

$$\bar{\mathbf{a}} \triangleq \mathbf{F}\,\tilde{\mathbf{a}} \tag{59}$$

$$\bar{\mathbf{A}} \triangleq \mathbf{F}\,\tilde{\mathbf{A}}\,\mathbf{F}^{-1}. \tag{60}$$

Then (see [21])

$$\bar{\mathbf{A}} = \operatorname{diag}(\bar{\mathbf{a}}) \tag{61}$$

$$\mathbf{F}\,\tilde{\mathbf{A}}^H\,\mathbf{F}^{-1} = (\mathbf{F}\,\tilde{\mathbf{A}}\,\mathbf{F}^{-1})^H = \bar{\mathbf{A}}^H = \bar{\mathbf{A}}^*. \tag{62}$$

With (61) we see that the similarity transform (59) always diagonalizes any circulant matrix and therefore the eigenvalue decomposition (EVD) of a circulant matrix always has the form

$$\tilde{\mathbf{A}} = \mathbf{F}^{-1}\,\bar{\mathbf{A}}\,\mathbf{F}. \tag{63}$$