COMPARISON AND A THEORETICAL LINK BETWEEN TIME-DOMAIN AND FREQUENCY-DOMAIN BLIND SOURCE SEPARATION

Robert Aichner, Herbert Buchner, Walter Kellermann
Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg
{aichner,buchner,wk}@LNT.de

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 $s_q(n)$, $q = 1, \ldots, P$ of our separation problem are filtered by a multiple input and multiple output (MIMO) system before they are picked up by the sensors x_p, $p = 1, \ldots, P$. 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 blind source separation (BSS), we are interested in finding a corresponding demixing system, where the output signals $y_q(n)$, $q = 1, \ldots, P$ are described by

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

where

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

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

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

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.

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,
- Nonstationarity property by simultaneous diagonalization of short-time output correlation matrices at different time intervals, e.g., [3, 4].

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 [5, 6].

In the following, we present a more general class of algorithms based on a matrix formulation for convolutive mixtures 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 algorithms as special cases of our framework.

2. A GENERIC BLOCK TIME-DOMAIN BSS ALGORITHM

In order to rigorously introduce multiple *time lags* in the cost function below, we define the following $N \times L$ block output signal matrix by incremental shifts of each column by one sample

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

with m being the block time index, and $N \geq L$ being the block length. When combining all channels, this leads to

$$\mathbf{Y}(m) = [\mathbf{Y}_1(m), \cdots, \mathbf{Y}_P(m)]. \tag{3}$$

Furthermore we define the short-time correlation matrix

$$\mathbf{R}_{yy}(m) = \mathbf{Y}(m)^H \mathbf{Y}(m) \tag{4}$$

and introduce the following cost function as a generalization of [3]:

$$\mathcal{J}(m) = \sum_{i=0}^{m} \beta(i,m) \left\{ \log\det \operatorname{bdiag} \mathbf{R}_{yy}(i) - \log\det \mathbf{R}_{yy}(i) \right\}, \tag{5}$$

where β is a normalized window function $\left(\sum_{i=0}^{m} \beta(i,m) = 1\right)$ allowing 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). 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.Since we use the matrix formulation (3) for calculating the short-time correlation matrices $\mathbf{R}_{yy}(m)$, the cost function inherently includes all time-lags of all auto-correlations and cross-correlations of the BSS output signals. The cost function becomes zero, if and only if all block-offdiagonal elements of $\mathbf{R}_{yy}$, i.e., the *output cross-correlations over all time-lags*, vanish.

In [2] it was shown that after deriving the natural gradient of (5) we obtain the following update rule:

$$\Delta\mathbf{W}(m) = 4 \sum_{i=0}^{m} \beta(i,m) \mathbf{W} \left\{ \mathbf{R}_{yy} - \operatorname{bdiag} \mathbf{R}_{yy} \right\} \operatorname{bdiag}^{-1} \mathbf{R}_{yy}, \tag{6}$$

This work was partly supported by the ANITA project funded by the European Comission under contract IST-2001-34327.

where the block-time index m of the correlation matrix is omitted for simplicity. $\mathbf{W}$ is a $2LP \times LP$ matrix consisting of $2L \times L$ Sylvester submatrices, which contain the L unmixing filter weights (see [2] for details).

To analyze the generalized update (6), and to study links to some known algorithms, we consider now the case $P = 2$ for simplicity. In this case, we have

$$\Delta\mathbf{W}(m) = 4\sum_{i=0}^{m}\beta(i,m) \cdot \begin{bmatrix} \mathbf{W}_{12}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{11}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \\ \mathbf{W}_{22}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{21}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \end{bmatrix}, \qquad (7)$$

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

In [5, 6], a time-domain algorithm was presented that copes very well with reverberant acoustic environments. Although it was originally introduced as a heuristic extension of [3] incorporating several time lags, this algorithm can be directly obtained from (7) by approximating $\mathbf{R}_{y_qy_q}$ in (7) as diagonal matrices with their diagonals consisting of the output signal powers. Using this approximation, the remaining products of Sylvester matrices and Toeplitz matrices in the update equation (7) can be efficiently implemented by a (fast) convolution as was done in [6].

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, 4]. 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.

It was shown in [2] that the matrix formulation introduced for the time-domain allows a rigorous derivation of frequency-domain algorithms which can be linked explicitly with their time-domain counterparts. The derivation is based on the principle that the Toeplitz matrices $\mathbf{Y}$ in (5) can be expressed by circulant matrices which are then diagonalizable by the discrete Fourier transform. This can be efficiently realized by using the fast Fourier transform. Deriving the gradient of the frequency-domain cost function with respect to $\underline{\mathbf{W}}$ leads to the following update equation [2]:

$$\nabla_{\underline{\mathbf{W}}}\mathcal{J}(m) = 4\sum_{i=0}^{m}\beta(i,m)\mathbf{S}_{xy}\mathbf{L}(\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L})^{-1}\mathbf{L}^H \cdot\{\mathbf{S}_{yy} - \text{bdiag}\,\mathbf{S}_{yy}\}\,\mathbf{L}\text{bdiag}^{-1}\,(\mathbf{L}^H\mathbf{S}_{yy}\mathbf{L})\mathbf{L}^H, \qquad (8)$$

where $\mathbf{L}$ denotes a $4LP \times LP$ constraint matrix (see [2] for details) and

$$\mathbf{S}_{xy} = \mathbf{S}_{xx}\underline{\mathbf{W}}; \qquad \mathbf{S}_{yy} = \underline{\mathbf{W}}^H\mathbf{S}_{xx}\underline{\mathbf{W}}$$
$$\mathbf{S}_{xx} = (\mathbf{G}_{4LP\times 4LP})^H\,\underline{\mathbf{X}}^H\mathbf{G}_{4L\times 4L}\underline{\mathbf{X}}\mathbf{G}_{4LP\times 4LP}.$$

with $\mathbf{G}_{4LP\times 4LP}$ and $\mathbf{G}_{4L\times 4L}$ being window matrices of dimension $4LP \times 4LP$ and $4L \times 4L$ respectively. The matrices $\underline{\mathbf{X}}_p$ and $\underline{\mathbf{W}}_{pq}$ denote the frequency-domain reprensentation of the input signals x_p and the filter weights w_{pq}, respectively:

$$\underline{\mathbf{X}}_p(m) = \text{diag}\{\mathbf{F}_{4L\times 4L}[x_p(mL-3L), \dots, x_p(mL-1), x_p(mL), x_p(mL+1), \dots, x_p(mL+L-1)]^T\}, \qquad (9)$$

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 $x_p(n)$ by means of the Fourier matrix $\mathbf{F}_{4L\times 4L}$.

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

Two types of constraints appear in the gradient (8):

- Matrix $\mathbf{L}$ introduces joint diagonalization over all time-lags.
- The matrices $\mathbf{G}_{\dots}$ 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 (8) we obtain a simplified algorithm utilizing only the nonstationarity of the source signals.

By additionally removing the constraints $\mathbf{G}_{\dots}$, i.e., approximating $\mathbf{G}_{\dots}$ by scaled identity matrices, all the submatrices in (8) become diagonal matrices. Only in this case (8) can be decomposed into independent update equations for each frequency bin $\nu = 0, \dots, 4L-1$. With an additional approximation, this update equation corresponds to that derived in [4]. In contrast to $\mathbf{S}_{xy}$ and $\mathbf{S}_{yy}$ in (8) which are $4LP \times 4LP$ matrices each, the corresponding matrices $\mathbf{S}_{xy}^{(\nu)}$ and $\mathbf{S}_{yy}^{(\nu)}$ are only of dimension $P \times P$. While this is computationally more efficient than (8), the known measures (e.g. [4]) to avoid internal permutation have to be taken.

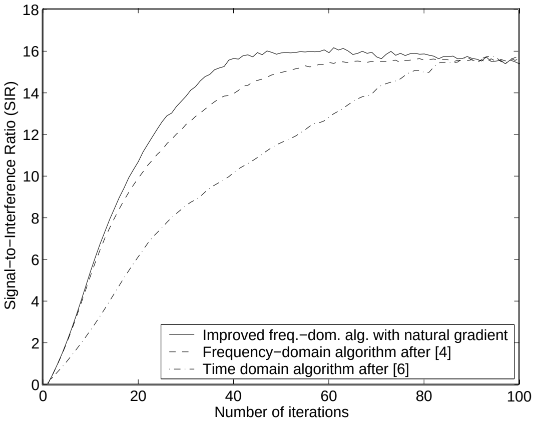

Fig. 1. Comparison of off-line implementations of the approximated versions of (7) and (8) for the 2×2 BSS model ($L = 512$ taps). Reverberation time $T_{60} = 150$ ms

4. REFERENCES

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

[2] H. Buchner, R. Aichner and W. Kellermann, “A Generalization of a class of blind source separation algorithms for convolutive mixtures,” in *Proc. ICA*, 2003.

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

[4] C.L. Fancourt and L. Parra, “The coherence function in blind source separation of convolutive mixtures of non-stationary signals,” in *Proc. NNSP*, 2001.

[5] 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.

[6] R. Aichner et al., “Time-domain blind source separation of non-stationary convolved signals with utilization of geometric beamforming,” in *Proc. NNSP*, pp. 445-454, 2002.