Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Signal Processing 86 (2006) 1260–1277

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

A real-time blind source separation scheme and its application to reverberant and noisy acoustic environments☆

Robert Aichner*, Herbert Buchner, Fei Yan, Walter Kellermann

Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, Cauerstr. 7, 91058 Erlangen, Germany

Received 31 January 2005; received in revised form 21 March 2005; accepted 28 June 2005
Available online 21 October 2005

Abstract

In this paper, we present an efficient real-time implementation of a broadband algorithm for blind source separation (BSS) of convolutive mixtures. A recently introduced generic BSS framework based on a matrix formulation allows simultaneous exploitation of nonwhiteness and nonstationarity of the source signals using second-order statistics. We demonstrate here that this general scheme leads to highly efficient real-time algorithms based on block-online adaptation suitable for ordinary PC platforms. Moreover, we investigate the problem of incorporating noncausal delays which are necessary with certain geometric constellations. Furthermore, the robustness against diffuse background noise, e.g., in a car environment is examined and a stepsize control is proposed which further enhances the robustness in real-world environments and leads to an improvement in separation performance. The algorithms were investigated in a reverberant office room and in noisy car environments verifying that the proposed method ensures high separation performance in realistic scenarios.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Blind source separation; Convolutive mixtures; Second-order statistics; Real-time; Stepsize control; Noisy BSS

1. Introduction

Blind source separation (BSS) refers to the problem of recovering signals from several observed linear mixtures. In acoustical scenarios the problem is referred to as the so-called cocktail party problem, where individual speech signals should be extracted from mixtures of multiple speakers in a usually reverberant acoustic environment. Due to the reverberation, the original source signals $s_q(n)$, $q = 1, \ldots, Q$ of our separation problem are filtered by a linear multiple-input and multiple-output (MIMO) system before they are picked up by the sensors. Moreover, in most environments (possibly spatiotemporally correlated) noise n_p as, e.g., sensor or background noise will be picked up by each sensor x_p, $p = 1, \ldots, P$. In the following, we assume that the number Q of source signals $s_q(n)$ (which may or may not be simultaneously active at a particular instant of time) equals the number of sensor signals $x_p(n)$, $p = 1, \ldots, P$. Thus, by the number of sensors we determine the maximum number of *simultaneously active* sources as a condition for perfect separation. Such scenarios

☆This work was partially supported by the ANITA project funded by the European Commission under Contract IST-2001-34327.

*Corresponding author.

E-mail address: aichner@LNT.de (R. Aichner).

0165-1684/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2005.06.022

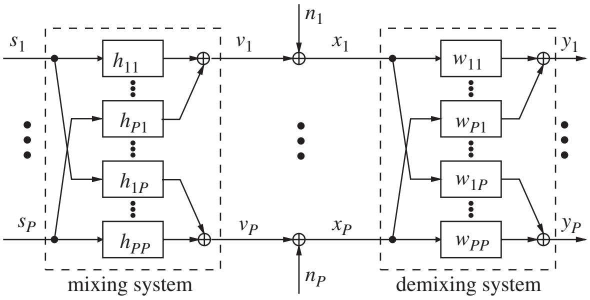

Fig. 1. Noisy BSS model.

are encountered, e.g., in speech acquisition in car environments as discussed later in Section 7 or in meeting rooms. An M-tap mixing system is thus described by

$$\begin{aligned} x_p(n) &= v_p(n) + n_p(n) \\ &= \sum_{q=1}^{P} \sum_{\kappa=0}^{M-1} h_{qp,\kappa} s_q(n-\kappa) + n_p(n), \end{aligned} \tag{1}$$

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

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

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

and where $w_{pq,\kappa}$, $\kappa = 0, \ldots, L-1$ denote the current weights of the MIMO filter taps from the pth sensor channel to the qth output channel. BSS is solely based on the fundamental assumption of mutual statistical independence of the different source signals. The separation is achieved by forcing the output signals y_q to be mutually statistically decoupled up to joint moments of a certain order. For convolutive mixtures as given by (1), frequency-domain BSS is very popular since all techniques originally developed for instantaneous BSS (i.e., for the special case $M = 1$ in (1)) may be applied independently in each frequency bin. This bin-wise processing, implying a narrowband signal model is denoted here as *narrowband approach* and is described, e.g., in [1]. Unfortunately, this traditional narrowband approach exhibits several limitations because ambiguities such as arbitrary permutation and scaling of the output signals may then also appear independently in each frequency bin. These ambiguities and also circular convolution effects have to be resolved by additional repair measures as, e.g., shown in [2,3]. In [4] a class of *broadband* algorithms was derived, for both the time domain and frequency domain, i.e., the frequency bins are no longer considered to be independent for real-world time-domain signals. These algorithms are based on *second-order statistics* simultaneously exploiting nonwhiteness and nonstationarity and inherently avoiding the permutation problem and circular convolution effects. Thus, no geometric information about the placement of the sensors is needed. In [5] this concept was also extended to higher-order statistics.

Furthermore, in [4] it has been shown that the optimum broadband BSS solution cancels the components from source s_p to all outputs y_q, $p \neq q$. This leads to an overall system $c_{pq,\kappa} = \sum_{j=1}^{P} \sum_{v=0}^{M-1} h_{pj,v} w_{jq,\kappa-v}$ where all cross-terms are cancelled, i.e., $c_{pq,\kappa} = 0$, $p \neq q$. In [6] a set of equations was formulated based on the overall system $c_{pq,\kappa}$ which resembles the conditions used in single-input multiple-output (SIMO) blind system identification [7]. Using this link it could be shown in [6] that the broadband BSS approaches actually perform *blind MIMO system identification* and thus, for a suitable choice of the demixing filter length L, avoid the filtering ambiguity (only an arbitrary scaling is possible). Traditionally, also multichannel blind deconvolution (MCBD) algorithms are applied to the BSS problem (e.g., [2,8]). There, due to the deconvolution, a temporal whitening of the output signals is observed in addition to the arbitrary filtering. Due to the system identification of our broadband approach, rather than a temporal deconvolution in MCBD-based BSS approaches, such whitening is also prevented.

In this paper we propose an efficient realization of one of these broadband algorithms based on second-order statistics which has led to a robust real-time implementation whereas current state-of-the-art real-time implementations are based on narrowband frequency-domain algorithms as, e.g., [9,10]. Based on the theoretical results in [11] we present two implementations of the efficient broadband algorithm. The first one uses causal demixing filters which are sufficient for several applications and proved to be very robust. The second implementation is more general and allows the adaptation of noncausal filters. As shown later, the necessity of noncausal filters arises in certain geometric constellations. In a practical BSS system another important aspect is its performance in noisy environments [1,12] which can be improved by suitable stepsize control techniques (e.g., [13]). In this contribution, we present a combination of online and offline processing which is termed block-online adaptation and investigate the influence of noise considering a theoretical link to a narrowband cost function based on the generalized coherence [14]. We also propose a suitable stepsize control which improves convergence and is robust to varying reverberance and noise conditions. Finally, the efficiency of this algorithm is demonstrated by experimental results for reverberant rooms and car environments with background noise for different geometric setups.

2. Generic block time-domain BSS algorithm

In [4,5,15] a generic BSS framework for convolutive mixtures has been presented. The real-time implementation presented in this paper is based on an efficient version derived from a special case of this framework based on second-order statistics and thus we briefly review this second-order case in Sections 2.1 and 2.2. In Section 2.3 a new iterative procedure based on a combination of offline and online updates to a so-called block-online algorithm is presented.

2.1. Matrix formulation

A block processing broadband algorithm simultaneously exploiting nonwhiteness and nonstationarity of the source signals is obtained by the following matrix formulation [15]. First, we introduce a block output signal matrix

$$\mathbf{Y}_q(m) = \begin{bmatrix} y_q(mL) & \cdots & y_q(mL-L+1) \\ y_q(mL+1) & \ddots & y_q(mL-L+2) \\ \vdots & \ddots & \vdots \\ y_q(mL+N-1) & \cdots & y_q(mL-L+N) \end{bmatrix}, \tag{3}$$

and reformulate the convolution (2) as

$$\mathbf{Y}_q(m) = \sum_{p=1}^{P} \mathbf{X}_p(m)\mathbf{W}_{pq}, \tag{4}$$

with m being the block time index and N denoting the block length. The $N \times L$ matrix $\mathbf{Y}_q(m)$ incorporates L *time-lags* into the correlation matrices in the cost function defined in Section 2.2, as is necessary for the exploitation of the nonwhiteness property. To ensure linear convolutions for all elements of $\mathbf{Y}_q(m)$, the $N \times 2L$ matrices $\mathbf{X}_p(m)$ and $2L \times L$ matrices $\mathbf{W}_{pq}$ are given as

$$\mathbf{X}_p(m) = \begin{bmatrix} x_p(mL) & \cdots & x_p(mL-2L+1) \\ x_p(mL+1) & \ddots & x_p(mL-2L+2) \\ \vdots & \ddots & \vdots \\ x_p(mL+N-1) & \cdots & x_p(mL-2L+N) \end{bmatrix}, \tag{5}$$

$$\mathbf{W}_{pq} = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{pq,0} & \ddots & \vdots \\ \vdots & w_{pq,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{pq,0} \\ 0 & w_{pq,L-1} & \ddots & w_{pq,1} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{pq,L-1} \\ 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{6}$$

where the matrices $\mathbf{X}_p(m)$, $p = 1, \ldots, P$ in (4) are Toeplitz matrices due to the shift of subsequent rows by one sample each. The matrices $\mathbf{W}_{pq}$ exhibit a Sylvester structure, where each column is shifted by one sample containing the current weights $\mathbf{w}_{pq} = [w_{pq,0}, w_{pq,1}, \ldots, w_{pq,L-1}]^{\mathrm{T}}$ of the MIMO sub-filter of length L from the pth sensor channel to the qth output channel. Superscript $^{\mathrm{T}}$ denotes transposition of a vector or a matrix. To allow a convenient notation of the algorithm combining all channels, we write (4) for all channels simultaneously as

$$\mathbf{Y}(m) = \mathbf{X}(m)\mathbf{W}, \tag{7}$$

with the matrices

$$\mathbf{Y}(m) = [\mathbf{Y}_1(m), \ldots, \mathbf{Y}_P(m)], \tag{8}$$

$$\mathbf{X}(m) = [\mathbf{X}_1(m), \ldots, \mathbf{X}_P(m)], \tag{9}$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{P1} & \cdots & \mathbf{W}_{PP} \end{bmatrix}. \tag{10}$$

2.2. Optimization criterion

The definition of $\mathbf{Y}_q$ in (3) leads to the short-time correlation matrix $\mathbf{R}_{\mathbf{yy}}(m) = \mathbf{Y}^{\mathrm{H}}(m)\mathbf{Y}(m)$ of size $PL \times PL$ which is composed of channel-wise $L \times L$ submatrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(m) = \mathbf{Y}_p^{\mathrm{H}}(m)\mathbf{Y}_q(m)$ each containing L time-lags. Based on these correlation matrices we use a cost function first introduced in [15] which inherently includes all L time-lags of all auto-correlations and cross-correlations of the BSS output signals and thus is a generalization of [16]:

$$\mathscr{J}(m,\mathbf{W}) = \sum_{i=0}^{\infty} \beta(i,m)\{\log \det \operatorname{bdiag} \mathbf{Y}^{\mathrm{H}}(i)\mathbf{Y}(i) - \log \det \mathbf{Y}^{\mathrm{H}}(i)\mathbf{Y}(i)\}. \tag{11}$$

Here $^{\mathrm{H}}$ denotes conjugate transposition and β is a weighting function with finite support that is normalized according to $\sum_{i=0}^{m} \beta(i,m) = 1$ allowing offline, online or block-online (see Section 2.3) realizations of the algorithm. In Section 2.3, it is shown that for a properly chosen $\beta(i,m)$ the nonstationarity of the source signals is utilized for the separation. 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 and are of size $L \times L$. The cost function becomes zero if and only if all block-offdiagonal elements of $\mathbf{Y}^{\mathrm{H}}\mathbf{Y}$, i.e., the *output cross-correlations over all time-lags*, become zero (see Fig. 2). Therefore, in addition to the nonstationarity, (11) explicitly exploits the nonwhiteness property of the output signals. In [5] it was shown that the cost function (11) also follows from an information-theoretic approach aiming at minimum mutual information between the output channels.

In order to express the update equations of the filter coefficients exclusively by Sylvester matrices $\mathbf{W}$, we take the generalization in [4] of the natural gradient [17] with respect to $\mathbf{W}$

$$\nabla_{\mathbf{W}}^{\mathrm{NG}} \mathscr{J}(m,\mathbf{W}) = 2\mathbf{W}\mathbf{W}^{\mathrm{H}} \frac{\partial \mathscr{J}(m,\mathbf{W})}{\partial \mathbf{W}^*}, \tag{12}$$

and ensure the Sylvester structure of the result by selecting only the non-redundant values using a *S*ylvester *C*onstraint (*SC*). A detailed discussion of *SC* will be given in step 3 of Section 3. In [4,15] it was shown that the natural gradient derivation of (11) leads to

$$\nabla_{\mathbf{W}}^{\mathrm{NG}} \mathscr{J}(m,\mathbf{W}) = 2 \sum_{i=0}^{\infty} \beta(i,m)\mathscr{Q}(i,\mathbf{W}), \tag{13}$$

$$\mathscr{Q}(i,\mathbf{W}) = \mathbf{W}\{\mathbf{R}_{\mathbf{yy}}(i) - \operatorname{bdiag} \mathbf{R}_{\mathbf{yy}}(i)\} \operatorname{bdiag}^{-1} \mathbf{R}_{\mathbf{yy}}(i). \tag{14}$$

Note that the submatrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}$ in (14) have to be properly regularized prior to inversion. Regularization strategies and two simple ways to impose the Sylvester constraint *SC* for the natural gradient are discussed in Section 3. In the following we will discuss how to obtain an iterative optimization procedure from the natural gradient (13), (14).

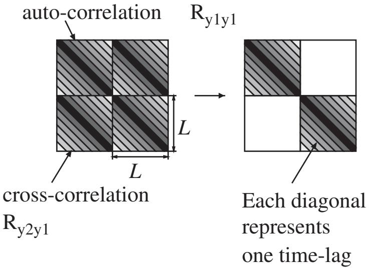

Fig. 2. Illustration of (11) for the 2 × 2 case.

2.3. Coefficient update rule

The weighting function $\beta(i,m)$ in (13) allows for different iterative optimization procedures of the algorithm, e.g., offline or online [5]. The concept of a general weighting function is already well-known from supervised adaptive filtering [18]. There, the weighting function $\beta(i,m) = (1-\lambda)\lambda^{m-i}\varepsilon_{0,m}(i)$ allows to derive the recursive least-squares (RLS) algorithm, i.e., the online solution, from the corresponding offline least-squares (LS) solution. The parameter λ denotes the exponential forgetting factor $(0<\lambda<1)$ and $\varepsilon_{a,b}(i)$ is a rectangular window function, i.e., $\varepsilon_{a,b}(i) = 1$ for $a \leqslant i \leqslant b$ and $\varepsilon_{a,b}(i) = 0$ elsewhere. In this paper, we want to use the same methodology to obtain a *recursive* block-by-block solution based on the offline minimization (were all data is required) by natural gradient descent given by

$$\mathbf{W}^j(m) = \mathbf{W}^{j-1}(m) - \tilde{\mu}\Delta\mathbf{W}^j(m), \quad j = 1,\ldots,j_{\max}, \tag{15}$$

where j denotes the iteration number, $\tilde{\mu}$ is the stepsize and the update $\Delta\mathbf{W}^j(m)$ corresponds to the natural gradient $\nabla_{\mathbf{W}}^{\mathrm{NG}} \mathscr{J}(m,\mathbf{W}^j(m))$ given in (13) together with some Sylvester constraint, discussed in Section 3. Here, the weighting function $\beta(i,m)$ is

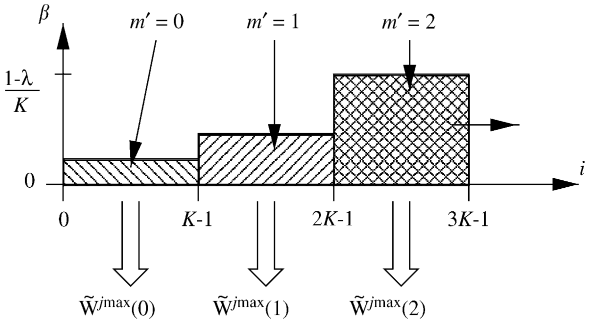
β
$m' = 0$
$m' = 1$
$m' = 2$
$\frac{1-\lambda}{K}$
0
i
0
K-1
$2K$-1
$3K$-1
$\tilde{\mathbf{W}}^{j_{\max}}(0)$
$\tilde{\mathbf{W}}^{j_{\max}}(1)$
$\tilde{\mathbf{W}}^{j_{\max}}(2)$

Fig. 3. Weighting function $\beta(i,m)$ for block-online implementation depicted for choosing the current composite block $m = 2$.

chosen as

$$\beta(i,m) = \frac{1-\lambda}{K} \sum_{m'=0}^{m} \lambda^{m-m'} \varepsilon_{m'K,m'K+K-1}(i), \tag{16}$$

and is shown in Fig. 3. The horizontal axis shows the block index i with each block having a length of N samples. Note that in Sections 2.1 and 2.2 the variable m denoted the current block of length N. When specifying $\beta(i,m)$ as given in (16), the current block m has to be extended to contain K subsequent blocks of length N with a blockshift of L samples each to allow the exploitation of the nonstationarity. This leads to a length of the current block m of $KL + N - L$ samples and m is termed composite block. Moreover, also the summation index $m' = 0, 1, \ldots, m$ in (16) refers to the composite block. As shown in Appendix A we can derive an approximate recursive formulation of the offline update (15) by using the weighting function $\beta(i,m)$ given in (16). This leads to a so-called block-online method where an online update and an offline update are combined similar to the approach in [10]. The advantage of this approach is that it allows a faster convergence and better tracking behaviour at moderate computational complexity.

According to Appendix A, the *offline* part is calculated iteratively for the current composite block m without exploiting any previous composite blocks (see Fig. 3) as:

$$\tilde{\mathbf{W}}^{j}(m) = \tilde{\mathbf{W}}^{j-1}(m) - \mu \tilde{\mathcal{Q}}(m, \tilde{\mathbf{W}}^{j-1}(m)), \quad j = 1, \ldots, j_{\max}, \tag{17}$$

$$\tilde{\mathcal{Q}}(m, \tilde{\mathbf{W}}^{j-1}(m)) = \frac{1}{K} \sum_{i=mK}^{mK+K-1} \mathcal{Q}(i, \tilde{\mathbf{W}}^{j-1}(m)), \tag{18}$$

where the stepsize $\mu = 2(1-\lambda)\tilde{\mu}$ and $\tilde{\mathbf{W}}^{j}(m)$ is the demixing filter matrix after j iterations based on data of the mth composite block. Eq. (18) contains K update terms $\mathcal{Q}(i, \tilde{\mathbf{W}}^{j-1}(m))$ which are determined using (14). This simultaneous optimization for K blocks allows to exploit the nonstationarity of the source signals as for each block the source statistics change and thus new conditions are generated. A high number of iterations $j_{\max}$ allows a fast convergence of the natural gradient descent without introducing an additional algorithmic delay but at the cost of an increased computational complexity. In practice, the maximum number of iterations $j_{\max}$ is usually chosen to $5 \ldots 10$ iterations to keep the complexity at a moderate level. The applicable initialization of $\tilde{\mathbf{W}}^{j-1}(m)$ for $m = 0$ and $j = 1$ results from the chosen Sylvester constraint as discussed in steps 3 and 4 of Section 3. The demixing filter matrix $\tilde{\mathbf{W}}^{j_{\max}}(m)$ of the current block m which is obtained from the offline part after $j_{\max}$ iterations (see Fig. 3) is then used as input of the *online* part of the block-online algorithm which is written recursively as

$$\mathbf{W}(m) = \lambda \mathbf{W}(m-1) + (1-\lambda)\tilde{\mathbf{W}}^{j_{\max}}(m). \tag{19}$$

This yields the final demixing filter matrix $\mathbf{W}(m)$ of the current composite block m containing the filter weights $\mathbf{w}_{pq}(\mathrm{m})$ used for separation. The demixing filter weights $\mathbf{w}_{pq}(m)$ of the current block are then used as initial values for the offline algorithm (17) of the next block.

Analogously to supervised block-based adaptive filtering [19], the approach followed here can also be carried out with overlapping data blocks in both, the online and offline part to further increase the convergence rate and to reduce the signal delay. The overlap factors α_{off} for the offline part and α_{on} for the online part with $1 \leqslant \alpha_{\mathrm{off}}, \alpha_{\mathrm{on}} \leqslant L$ should be chosen suitably to obtain integer values for the time index.

3. Approximated version and efficient implementation

In this section, we address implementation details concerning the update term $\mathcal{Q}(i, \mathbf{W})$ of the ith block of length N in (18) which lead to an efficient implementation suitable for real-time processing. All steps are summarized in Table 1 where a pseudocode for the proposed algorithm is given.

Step 1: Estimation of the correlation matrices using the correlation method.

Table 1
Pseudo code of efficient block-online broadband algorithm implementation based on Sylvester constraint SC_C exemplarily shown for the update $\Delta\mathbf{w}_{11}(m)$ in the 2×2 case

```
Online part:
1.  Get $KL+N$ new samples $x_p(mKL),\ldots,x_p((m+1)KL+N-1)$
    of the sensors $x_p, p=1,2$ and online block index $m=0,1,2,\ldots$
Offline part:
    Compute for each iteration $j=1,\ldots,j_{\max}$:
2.      Compute output signals $y_q(mKL),\ldots,y_q((m+1)KL+N-L-1)$,
        $q=1,2$ by convolving $x_p$ with filter weights $\mathbf{w}_{pq}^{j-1}$from previous iteration
3.      Generate $K$ blocks of $N$ samples $[y_q(iL),\ldots,y_q(iL+N-1)]$ with off-
        line block index $i=mK,\ldots,mK+K-1$ to exploit nonstationarity
        Compute for each block $i=mK,\ldots,mK+K-1$:
4.          Calculate the signal energy of each block$m$
            $\sigma_{y_1}^2(i) = r_{y_1y_1}(i,0) = \sum_{n=iL}^{iL+N-1} y_1^2(n)$
5.          Compute 1st column of cross-correlation matrix $\mathbf{R}_{\mathbf{y}_2\mathbf{y}_1}(i)$ by
            $r_{y_2y_1}(i,u)$ for $u=-L+1,\ldots,0$ according to (21)
6.          Normalize by elementwise division with regularized signal energy
            $r_{y_2y_1}(i,u)/(\sigma_{y_1}^2(i)+\delta_{y_1})$ for $u=-L+1,\ldots,0$
7.          Compute the matrix product $\tilde{\mathbf{W}}_{12}^{j-1}(m)\,\frac{\mathbf{R}_{\mathbf{y}_2\mathbf{y}_1}(i)}{\sigma_{y_1}^2(i)+\delta_{y_1}}$ as a convolution
            according to Fig. 4a(Sylvester constraint $SC_C$). Each filter weight
            update $\Delta w_{11,\kappa}^j$, $\kappa=0,\ldots,L-1$ is thus calculated as:
            $\tilde{\mathcal{Q}}(m,\tilde{\mathbf{W}}_{11}^{j-1}(m)) = \frac{1}{K}\sum_i \sum_{n=0}^{\kappa} w_{12,n}^{j-1}(m) r_{y_2y_1}(i,\kappa-n)/(\sigma_{y_1}^2(i)+\delta_{y_1})$
8.      Update equation for the offline part (Note that also an adaptive and
        DFT bin-wise stepsize according to Section 6 can be applied):
        $\tilde{\mathbf{W}}_{11}^{j}(m) = \tilde{\mathbf{W}}_{11}^{j-1}(m-1) - \mu\tilde{\mathcal{Q}}(m,\tilde{\mathbf{W}}_{11}^{j-1}(m))$
Onlinepart:
9.  Compute the recursive update of the online part yielding the demixing
    filter $\mathbf{W}_{11}(m)$ used for separation:
    $\mathbf{W}_{11}(m) = \lambda\mathbf{W}_{11}(m-1) + (1-\lambda)\tilde{\mathbf{W}}_{11}^{j_{\max}}(m)$
10. Compute Steps 4–9 analogously for the other channels and use the
    demixing filter $\mathbf{W}_{pq}(m)$ as the initial filter for the offline part
    $\tilde{\mathbf{W}}_{pq}^{0}(m+1) = \mathbf{W}_{pq}(m)$
```

In principle, there are two basic methods for the block-based estimation of the short-time output correlation matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i)$ for nonstationary signals: the so-called *covariance method* and the *correlation method*, as they are known from linear prediction problems [20].[1] In the generic framework in Sections 2.1 and 2.2 the more accurate covariance method was introduced by the definition $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i) = \mathbf{Y}_p^{\mathrm{H}}(i)\mathbf{Y}_q(i)$. By assuming stationarity within each block i we can approximate the covariance method by the correlation method which exhibits lower computational

[1] It should be emphasized that the terms *covariance method* and *correlation method* are *not* based upon the standard usage of the

(*footnote continued*)
covariance function as the correlation function with the means removed.

complexity. This leads to a *Toeplitz structure* of the $L \times L$ matrix $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i)$ which can be expressed as

$$\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}(i) = \begin{bmatrix} r_{y_py_q}(i,0) & \cdots & r_{y_py_q}(i,L-1) \\ r_{y_py_q}(i,-1) & \ddots & r_{y_py_q}(i,L-2) \\ \vdots & \ddots & \vdots \\ r_{y_py_q}(i,-L+1) & \cdots & r_{y_py_q}(i,0) \end{bmatrix}, \tag{20}$$

$$r_{y_py_q}(i,u) = \begin{cases} \sum_{n=iL}^{iL+N-u-1} y_p(n+u)y_q(n) & \text{for } u \geqslant 0 \\ \sum_{n=iL-u}^{iL+N-1} y_p(n+u)y_q(n) & \text{for } u < 0 \end{cases}. \tag{21}$$

Step 2: Approximation of the normalization and regularization strategy.

A straightforward implementation of (14) together with (20), (21) leads to a complexity of $\mathcal{O}(L^2)$ due to the inversion of P auto-correlation Toeplitz matrices $\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}$ of size $L \times L$ which are normalizing the update (as also known from the recursive least-squares (RLS) algorithm in supervised adaptive filtering [18]). Thus, for an efficient implementation suitable for reverberant environments requiring a large filter length L we use an approximated version of (14) which was first heuristically introduced in [21,22] and theoretically derived in [4]. The efficient version is obtained by approximating the auto-correlation submatrices in the normalization term by the output signal powers, i.e.,

$$\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}(i) \approx \left(\sum_{n=iL}^{iL+N-1} y_q^2(n) \right) \mathbf{I} = \sigma_{y_q}^2(i)\mathbf{I} \tag{22}$$

for $q = 1, \ldots, P$. Thus, the matrix inversion is replaced by an element-wise division. This is comparable to the normalization in the well-known normalized least mean squares (NLMS) algorithm in supervised adaptive filtering approximating the RLS algorithm [18].

In blocks with speech pauses and low background noise the output powers $\sigma_{y_q}^2$ are very small and thus the parameter estimation becomes very sensitive. For a robust adaptation $\sigma_{y_q}^2$ is replaced by a regularized version $\sigma_{y_q}^2 + \delta_{y_q}$. The basic feature of the regularization is a compromise between fidelity to data and fidelity to prior information about the solution [23]. As the latter increases robustness, but leads to biased solutions, we use similarly to supervised adaptive filtering [24] a dynamical regularization

$$\delta_{y_q} = \delta_{\max} \mathrm{e}^{-\sigma_{y_q}^2/\sigma_0^2} \tag{23}$$

with two parameters $\delta_{\max}$ and σ_0^2. This exponential method provides a smooth transition between regularization for low output power $\sigma_{y_q}^2$ and data fidelity whenever the output power is large enough.

Step 3: Efficient implementation of the matrix–matrix multiplication.

In the remaining channel-wise matrix product of $\mathbf{W}_{pt}$ and the Toeplitz matrices $\mathbf{R}_{\mathbf{y}_t\mathbf{y}_q}/(\sigma_{y_q}^2 + \delta_{y_q})$, $p, q, t = 1, \ldots, P$, $t \neq q$ in the filter update $\mathcal{Q}(i)$ in (14) we can exploit the Sylvester structure of $\mathbf{W}_{pt}$ for an efficient implementation. As already mentioned in Section 2.2, we have to ensure by a suitable Sylvester constraint SC that the update $\mathcal{Q}(i, \mathbf{W})$ exhibits again a channel-wise Sylvester structure in the form of (6). As discussed in [11] there are two simple realizations of SC leading to different filter updates $\Delta w_{pq,\kappa}$ of the FIR filter taps $w_{pq,\kappa}$ and thus to two algorithms with different properties:

(1) Computing only the *first column* of the update matrix $\Delta\mathbf{W}_{pq}$ (whose structure is defined analogously to (6)) and replicating these elements to obtain a Sylvester structure. This is denoted as SC_C and is illustrated in Fig. 4a. Note that in general, when exploiting more than L time-lags, any column could be chosen. However, in [4, Section II-D] it was shown that the most efficient version is obtained by choosing the first column.

(2) Computing only the *Lth row* of the update matrix $\Delta\mathbf{W}_{pq}$ and replicating the filter weights is denoted SC_R (Fig. 4b). It should be noted that when implementing SC row-based, the Lth row has to be chosen as only this row contains all filter updates $\Delta w_{pq,\kappa}$, $\kappa = 0, \ldots, L-1$.

It can be seen that for both, SC_C and SC_R the matrix-matrix multiplication boils down to a matrix-vector product. Furthermore, it can be shown that due to the structure of the respective matrix these products denote linear convolutions of the filter weights $\mathbf{w}_{pt}$ with the elements of the scaled Toeplitz correlation matrix $\mathbf{R}_{\mathbf{y}_t\mathbf{y}_q}/(\sigma_{y_q}^2 + \delta_{y_q})$. Depending on the chosen Sylvester constraint different correlation sequences are used for the convolution. The version using SC_C convolves the filter weights

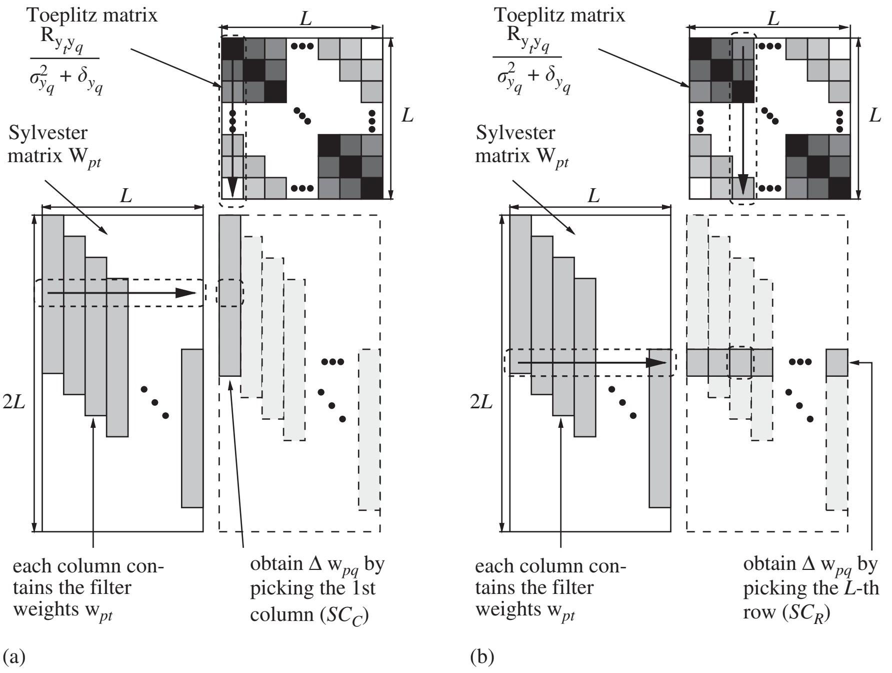
Toeplitz matrix
$\frac{\mathbf{R}_{y_t y_q}}{\sigma^2_{y_q}+\delta_{y_q}}$
L
L
Sylvester matrix $\mathbf{W}_{pt}$
L
2L
each column contains the filter weights $\mathbf{w}_{pt}$
obtain Δ $\mathbf{w}_{pq}$ by picking the 1st column (SC_C)
(a)
Toeplitz matrix
$\frac{\mathbf{R}_{y_t y_q}}{\sigma^2_{y_q}+\delta_{y_q}}$
L
L
Sylvester matrix $\mathbf{W}_{pt}$
L
2L
each column contains the filter weights $\mathbf{w}_{pt}$
obtain Δ $\mathbf{w}_{pq}$ by picking the L-th row (SC_R)
(b)

Fig. 4. Illustration of the channel-wise matrix–matrix product when using the (a) Sylvester Constraint SC_C and (b) Sylvester constraint SC_R.

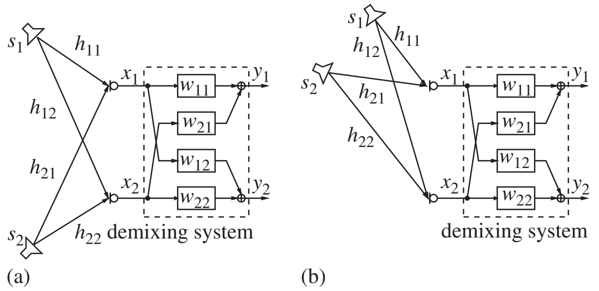

Fig. 5. Setups for BSS requiring: (a) only causal delays and (b) causal and noncausal delays for the demixing system $\mathbf{W}$.

with the one-sided sequence of correlation elements $r_{y_t y_q}(i,u)$, $u = 0, \ldots, -L+1$ whereas SC_R allows to use the two-sided sequence $r_{y_t y_q}(i,u)$, $u = -L+1, \ldots, L-1$. These two implementation schemes have an effect on the properties of the resulting algorithm (see experiments in Section 7) and on the suitable initialization of $\mathbf{W}_{pq}$ as will be discussed in the following. Moreover, by implementing the linear convolution expressed by the matrix-vector product as a fast convolution using fast Fourier transforms (FFTs) the computational complexity can be reduced to $\mathcal{O}(\log L)$.

Step 4: Appropriate initialization.

Depending on the BSS setup different initialization methods are desirable. This can be seen when regarding BSS as a blind interference cancellation technique similarly to conventional adaptive beamforming: In the acoustic scenario in Fig. 5a causal filters are sufficient to achieve interference cancellation and thus an initialization with a *unit impulse at the first tap* $w_{pp,0} = 1$ is sufficient. However, for the source locations in Fig. 5b in the BSS case one noncausal demixing filter w_{12} or w_{21} is required. Similarly to the supervised filtering algorithms used in adaptive beamforming [25] the problem of noncausality can be solved in the BSS context by initializing the FIR filters $w_{pp,\kappa}$ with a *shifted unit impulse*. An appropriate shift is determined by the array geometry and the resulting maximum possible delay of the arriving signals between the sensors. In [11] the shift of the unit impulse was set to $L/2$.

The choice of initialization method also determines the suitable Sylvester constraint SC. In the case of causal mixtures (Fig. 5a), i.e., initialization with $w_{pp,0} = 1$ both Sylvester constraints SC_C and SC_R are possible. Our experiments showed that algorithms based on SC_C are slightly more robust.

As described above, for noncausal mixtures it is necessary to initialize with a shifted unit impulse as, e.g., $w_{pp,L/2}=1$. When evaluating the matrix product resulting from SC_C (Fig. 4a) for such an initialization it can be seen that all $\Delta w_{pq,\kappa}$ for $0\leqslant\kappa\leqslant L/2-1$ would be equal to zero, i.e., these filter coefficients could not be adapted. Thus, for the initialization with a shifted unit impulse only SC_R can be applied. Then, similarly to adaptive beamforming also noncausal delays are possible as becomes obvious by simply evaluating (6) for successive iterations. A more detailed examination of the Sylvester constraints SC_C and SC_R and the resulting links between the generic update (13), (14) and various MCBD algorithms can be found in [11].

It can be concluded that if no a priori information about the location of the sources is available then the Sylvester constraint SC_R together with an initialization using a shifted unit impulse should be applied due to its increased generality. If it is known that the sources are in two halfplanes (e.g., car environment with array at the rear mirror), then also the more robust SC_C can be used. In Table 1 the algorithm has been summarized and the pseudocode is given exemplarily for the update $\Delta\mathbf{w}_{11}(m)$. As mentioned in Section 2.3, overlap factors can be introduced in the offline and online part. This is done by simply replacing the time index iL and mKL in Table 1 by iL/α_{off} and mKL/α_{on}, respectively.

4. Computational complexity

The computational complexity of the proposed algorithm is studied in terms of arithmetic operations, i.e., the number of real multiplications and real additions. Divisions are usually counted as multiplications, assuming inverted constants and subtraction is addition by negated number. Thereby, each complex multiplication is realized by 4 real multiplications and 2 real additions and each complex addition is realized by 2 real additions. Moreover, the discrete Fourier transform of length N is computed using the FFT routine devised by [26] which requires $2N\log_2[N]-(3N/2)-4$ operations.

Table 2 shows the computational complexity of the block-online broadband algorithm as presented

Table 2
Computational complexity for the block-online broadband algorithm implementation

	Arithmetic OPs for K blocks and P channels
Compute offline part for each iteration j:	
Perform filtering of x_p with $\mathbf{W}^j_{pq}(m')$:	
FFT of demixing filter with FFT length $KL+N$	$P^2(2(KL+N)\log_2[KL+N]-3(KL+N)/2-4)$
FFT of sensor signals x_p	$P(2(KL+N)\log_2[KL+N]-3(KL+N)/2-4)$
Compute convolution in DFT domain	$(4P^2+P)(KL+N+2)$
IFFT to obtain time-domain signals y_q	$P(2(KL+N)\log_2[KL+N]-3(KL+N)/2-4)$
Compute for each block $i=1,\ldots,K$:	
Calculate scaling factor $\sigma^2_{y_q}(i)$ after (22)	$P(N+(K-1)(4L+2))$
Calculate cross-correlations $r_{y_py_q}(i,u)$ for $u=-L+1,\ldots,L-1$:	
FFT of output signals y_q with length $2N$	$PK(4N\log_2[2N]-3N-4)$
Compute cross-power spectral densities	$3(P^2-P)K(N+1)$
IFFT to obtain cross-correlations	$(P^2-P)K(4N\log_2[2N]-3N-4)/2$
normalizer$r_{y_py_q}(i,u)$ using $\sigma^2_{y_q}(i)$	$2(P^2-P)KL$
Calculate matrix product as convolution:	
FFT of demixing filters $w_{pq,\kappa}$ of length $2L$	$P^2(4L\log_2[2L]-3L-4)$
FFT of cross-correlations of length $2L$	$(P^2-P)K(4L\log_2[2L]-3L-4)/2$
Compute convolution in DFT domain	$P^2(8P-10)K(L+1)$
IFFT	$P^2K(4L\log_2[2L]-3L-4)$
add $\mathcal{Q}(i,\tilde{\mathbf{W}}^{j-1})$ for K blocks after (18)	$P^2(K-1)L$
offline update rule (17)	$2P^2L$
online update rule (19)	$3P^2L$

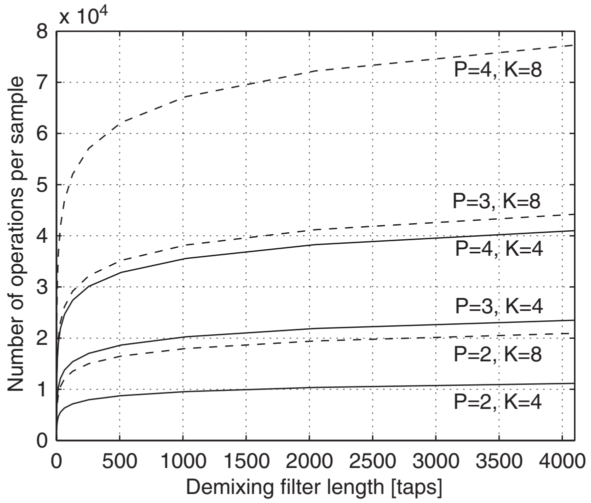

Fig. 6. Computational complexity for various filter lengths L and number of channels ($P = 2, 3, 4$) for simultaneous processing of $K = 4$ (solid) and $K = 8$ (dashed) blocks, respectively.

in Section 3 and summarized in Table 1. In Fig. 6, the complexity resulting from the expressions in Table 2 is illustrated as a function of filter length L and for $P = 2, 3, 4$ source and sensor signals. Additionally, the dependency of the complexity on the number K of simultaneously processed blocks to exploit the nonstationarity is illustrated by comparing $K = 4$ (solid) to $K = 8$ (dashed) in Fig. 6. The overlap factor of the online part has been chosen as $\alpha_{\text{on}} = K$ to ensure a blockshift of L samples independent of the choice of K. The number of iterations and the block length have been chosen as in the simulations below: $j_{\max} = 5, N = 2L$. The curves illustrate that, essentially, the complexity depends logarithmically on the filter length L, linearly on the number of blocks K, but quadratically on the number of channels P. For comparison it should be noted that the well-known (single-channel) NLMS algorithm used in supervised adaptive filtering [18] has a complexity of $4L + 7$ arithmetic operations. Thus, for example, the complexity of the two-channel BSS algorithm for $K = 4$ and $L \approx 2000$ corresponds, according to Fig. 6 approximately to that of a single-channel NLMS algorithm for the same filter length.

5. Performance in noisy environments

In Section 2.2, it was pointed out that the broadband algorithm minimizes the cross-correlations $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_q}$, $p \neq q$ for all time lags (see also Fig. 2). In [4,5,15] it was shown that the broadband algorithm and thus also the broadband cost function (11) can be formulated equivalently in the frequency-domain. By such a formulation there are still linear convolutions computed, so that the frequency bins are still linked together by certain constraint matrices and thus the ambiguities of the narrowband approaches mentioned in Section 1 are still avoided. In [15] it was shown that the broadband algorithm is minimizing the cross-power spectral densities $\mathbf{S}_{\mathbf{y}_p\mathbf{y}_q}^{(\nu)}$, $p \neq q$ for each frequency bin $\nu = 0, \ldots, 2L - 1$. Therefore, for the purpose of examining the noise-robustness of the proposed broadband algorithm with respect to the frequency domain we can, similarly to [15] and without loss of generality, express the cost function with respect to the νth frequency bin

$$\mathcal{J}^{(\nu)}(m) = \sum_{i=0}^{m} \beta(i,m)\{\log\det\operatorname{diag}\mathbf{S}_{\mathbf{yy}}^{(\nu)}(i) - \log\det\mathbf{S}_{\mathbf{yy}}^{(\nu)}(i)\}, \tag{24}$$

where $\mathbf{S}_{\mathbf{yy}}^{(\nu)}$ is the $P \times P$ cross-power spectral density matrix in the νth frequency bin. Moreover, we can approximate (24) by a Taylor series as shown in [15], to obtain

$$\mathcal{J}^{(\nu)}(m) \approx \sum_{i=0}^{\infty} \beta(i,m)\left\{1 - \frac{\det\mathbf{S}_{\mathbf{yy}}^{(\nu)}(i)}{\prod_{p=1}^{P} S_{y_py_p}^{(\nu)}(i)}\right\}, \tag{25}$$

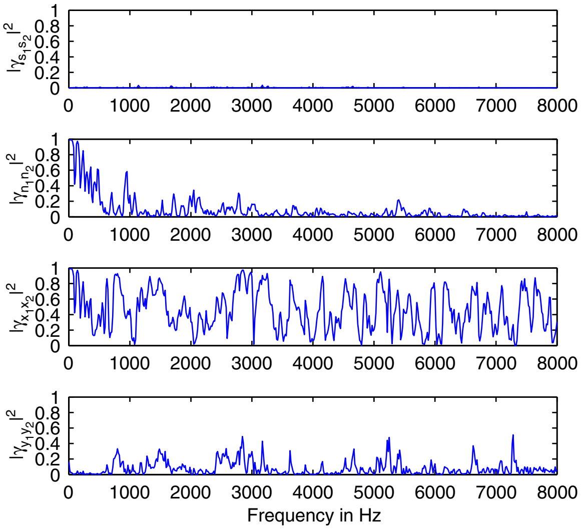

Fig. 7. Magnitude squared coherence function $|\gamma|^2$ of source signals, car noise, microphone and output signals (microphone spacing $d = 20$ cm).

where the term in brackets denotes the *generalized coherence* introduced in [14]. Note that for $P=2$ the generalized coherence corresponds to the well-known coherence $\gamma_{y_1y_2}$ between the output channels y_1 and y_2. It can be seen that the cost function (25) becomes zero if and only if the *cross*-power spectral densities of the output signals $S^{(v)}_{y_py_q}$, i.e., the off-diagonal elements of $\mathbf{S}^{(v)}_{\mathbf{yy}}$ are zero. Thus, the iterative broadband algorithm tries to minimize the coherence $\gamma_{y_1y_2}$ for all frequency bins v (see Fig. 7). To evaluate the influence of noise we express $\mathbf{S}^{(v)}_{\mathbf{yy}}$ in terms of $\mathbf{S}^{(v)}_{\mathbf{xx}}$ and decompose this matrix according to (1) into its speech signal and noise components (assuming orthogonality of speech and noise)

$$\mathbf{S}^{(v)}_{\mathbf{yy}} = \underline{\mathbf{W}}^{(v)^{\mathrm{H}}}\mathbf{S}^{(v)}_{\mathbf{xx}}\underline{\mathbf{W}}^{(v)} = \underline{\mathbf{W}}^{(v)^{\mathrm{H}}}(\mathbf{S}^{(v)}_{\mathbf{vv}} + \mathbf{S}^{(v)}_{\mathbf{nn}})\underline{\mathbf{W}}^{(v)}. \qquad (26)$$

For noise which is spatially uncorrelated (e.g., sensor noise), $\mathbf{S}^{(v)}_{\mathbf{nn}}$ corresponds to a diagonal matrix whereas for spatiotemporally correlated noise (e.g., diffuse noise) the matrix $\mathbf{S}^{(v)}_{\mathbf{nn}}$ is not sparse. In [6] it was shown that when choosing an appropriate demixing filter length L, arbitrary filtering by the broadband BSS algorithm is prevented. This means that the iterative algorithm mainly affects the cross-power spectral densities $S^{(v)}_{y_py_q}$. Thus, with the initialization discussed in Section 3 the diagonal noise term, i.e., the spatially uncorrelated noise of $\mathbf{S}^{(v)}_{\mathbf{nn}}$ of the initial block leads to a bias of $S^{(v)}_{y_py_p}$ which cannot be removed by the BSS algorithm. For a given SNR this bias will be more severe for spatially uncorrelated noise whereas for spatially correlated noise the initial bias is distributed among all elements of $\mathbf{S}^{(v)}_{\mathbf{yy}}$. It should be noted that the spatially correlated noise components appearing at the cross-power spectral densities $S^{(v)}_{y_py_q}$ will be minimized by the BSS algorithm leading to an SNR gain at the outputs. In Fig. 7 this becomes obvious when comparing $\gamma_{n_1n_2}$ and $\gamma_{y_1y_2}$.

To increase robustness of BSS algorithms against uncorrelated noise, bias removal techniques have been introduced (see, e.g., [1,12]), mainly consisting in the estimation and subtraction of an estimated diagonal, i.e., spatially uncorrelated noise matrix $\mathbf{S}^{(v)}_{\mathbf{nn}}$ from $\mathbf{S}^{(v)}_{\mathbf{yy}}$. To deal with spatiotemporally correlated and slowly time-varying noise, we propose to use the minimum statistics approach [27] for the estimation of the noise characteristics. This method is based on the observation that the power of a noisy speech signal frequently decays to the power of the background noise. Hence by tracking the minima we obtain an estimate for the auto-power spectral density of the noise. However, due to the spatial correlation not only the auto- but also the cross-power spectral densities of the noisy signal x_p and the background noise n_p are required. They are estimated and averaged recursively for each frequency bin whenever we detect a minimum (i.e., speech pause) of the noisy speech signals. Thus, for slowly time-varying noise statistics this method gives an accurate estimate of the noise spectral density matrix used for the bias removal. Note that for multiple active speakers this estimation problem is more difficult than for a single speaker due to fewer speech pauses.

6. Stepsize control

In general for adaptive algorithms the choice of the stepsize is important. In an offline processing scheme several trials can be run to maximize the stepsize up to the stability margin. However, for an online procedure usually the stepsize has to be chosen very conservatively to prevent instability problems. To make the adaptation more robust in real-world environments a stepsize control is desirable. In supervised adaptive filtering usually a closed-form solution for the stepsize is derived based on an observable reference signal. So far, in the BSS community there is little literature on this topic due to the absence of a reference signal. In the instantaneous mixing case some adaptive stepsize methods have been proposed (e.g., [13,28]) which are mainly relying on second order derivatives. However, for the convolutive mixing case such gradient stepsizes are computationally complex. In the neural networks community iterative methods for stepsize determination based on online measurements of the state of the adaptive system are more common and can be found in textbooks as, e.g., [23,29]. We propose to use a simple but effective strategy for updating the stepsize in our real-time BSS system based on a method presented in [23, p. 146, 30]. The procedure is to increase the stepsize if the value of the cost function $\mathcal{J}(m)$ is decreased compared to $\mathcal{J}(m-1)$ (indicating convergence) and to decrease it rapidly if the current value of $\mathcal{J}(m)$ exceeds the previous one $\mathcal{J}(m-1)$, by more than a prespecified ratio (indicating divergence). In the last case the current demixing filter update may be discarded ($\Delta\mathbf{W}(m)=0$). After starting with a small stepsize $\mu(0)$ its modifications are described by

$$\mu(m+1) = \begin{cases} a\cdot\mu(m) & \text{if } \mathscr{J}(m) < \mathscr{J}(m-1),\ a>1, \\ b\cdot\mu(m) & \text{if } \mathscr{J}(m) \geqslant c\cdot\mathscr{J}(m-1), \\ & \qquad b<1, c>1, \\ \mu(m) & \text{otherwise} \end{cases} \tag{27}$$

where for a wide range of applications the values $a = 1.1$, $b = 0.5$, $c = 1.3$ provided robust behavior. Moreover, to avoid instabilities, in practice the adaptive stepsize should be restricted to a finite range $[\mu_{\min}, \mu_{\max}]$. In (27) the cost function $\mathscr{J}$ given in (11) has to be evaluated for each block m. It can be seen in (11) that this involves the computation of the determinant of a large $PL \times PL$ matrix $\mathbf{Y}^{\mathrm{H}}\mathbf{Y}$. To avoid this additional computational complexity we use the already computed cross-correlation sequences $r_{y_p y_q}(m, u)$, $p \neq q$ instead. By taking the L_2-norm of the cross-correlation sequence a scalar value is obtained which is used to replace $\mathscr{J}$ for the decision process in (27).

In the preceding section the influence of noise was examined and it was shown in Fig. 7 that the noise characteristics change depending on the frequency region. Thus it is desirable to use a frequency-dependent adaptive stepsize $\mu^{(v)}(m)$. This stepsize can be calculated for every frequency bin $v = 0, \ldots, 2L-1$ according to the algorithm given in (27) where $\mathscr{J}$ will be replaced by the narrowband frequency-domain cost function $\mathscr{J}^{(v)}$. The advantage of using this bin-wise approach is that $\mathscr{J}^{(v)}$ can be approximated by the generalized coherence as shown in (25) which can be calculated with little additional computational complexity. For applying the bin-dependent stepsize, the update $\Delta\mathbf{W}(m)$ is transformed by an FFT, multiplied in each frequency bin by $\mu^{(v)}(m)$ and transformed back into the time-domain using an IFFT.

Furthermore, more sophisticated schemes which apply individual adaptive stepsizes to different filters are possible. This can be useful if, e.g., only one speaker is moving and thus, only a few demixing filters $\mathbf{W}_{pq}(m)$ are highly affected.

7. Experimental results

In this section, we present the experimental results for the proposed algorithm which is implemented in a real-time system on a standard PC platform. In a previous conference contribution [31] results for moving speakers were presented. Here, we want to first examine different stepsize control techniques (Section 6). Then, results for two reverberant enclosures and two geometric setups (see Fig. 5) are presented. They show the applicability of the two Sylvester constraints SC_C and SC_R with regard to causal and noncausal acoustical scenarios. Furthermore, experimental results in a car environment are given and the influence of the background noise is shown. In the end the output signal quality is examined.

The experiments have been conducted using speech data convolved with measured impulse responses of two rooms (Section 7.1) and a car environment (Section 7.2), respectively. A two-element microphone array with an inter-element spacing of 20 cm was used for the recordings ($P = 2$). Sentences spoken by a male and female speaker sampled at $f_s = 16$ kHz were selected as source signals ($Q = 2$). To evaluate the performance, the signal-to-interference ratio (SIR) averaged over both channels was calculated in each block which is defined as the ratio of the signal power of the *desired speaker* to the signal power from the *interfering speaker*.

7.1. Reverberant rooms

The impulse responses were measured in two rooms, a low reverberation chamber with a reverberation time $T_{60} = 50$ ms and an office room (580 cm × 590 cm × 310 cm), with $T_{60} = 250$ ms. The loudspeaker-microphone distance is 1 m for the low reverberation chamber and 2 m for the office room. For the first experiment, the evaluation of the proposed stepsize control, the speech signals arrived from -45° and $+45^\circ$. The parameters of the algorithm were chosen as $L = 1024$, $N = 2048$, $K = 8$, $\alpha_{\mathrm{on}} = 8$ resulting in an algorithmic delay of 1024 samples (64 ms). The latency of the system includes an additional hardware dependent delay of the audio interface which was about 25 ms in this system. The offline-part was calculated for $j_{\max} = 5$ iterations and for the online part the forgetting factor $\lambda = 0.2$ was chosen. According to Section 6 three different stepsize controls have been investigated:

(1) A fixed stepsize $\mu = 0.01$ which was maximized up to the stability margin.
(2) An adaptive stepsize $\mu(m)$ based on (27) identically chosen for all frequency bins. The

parameters have been chosen as $a = 1.1$, $b = 0.5$, $c = 1.3$, $\mu_{\min} = 0.0001$, $\mu_{\max} = 0.01$ and the initial value $\mu(0) = 0.01$.

(3) A bin-selective stepsize control $\mu^{(\nu)}(m)$ updated for each frequency bin $\nu = 0, \ldots, 2L-1$ according to the procedure in (27). The parameters are chosen analogously as for $\mu(m)$.

In Fig. 8 the experimental results for all three stepsize controls are shown exemplarily for the low reverberation chamber. It can be seen that the algorithm converges quickly due to the block-online structure and that an adaptive stepsize improves the convergence and also the maximum achievable SIR. This could be verified also for other enclosures and reverberation times. Moreover, the bin-selective stepsize control $\mu^{(\nu)}(m)$ outperforms the time-domain stepsize $\mu(m)$. Thus, in the following we use the bin-selective stepsize. A further advantage of the adaptive stepsizes is that they prevent divergence of the demixing filters in the real-time system. This allows the choice of a relatively large stepsize $\mu(m)$ or $\mu^{(\nu)}(m)$ compared to a fixed learning rate μ.

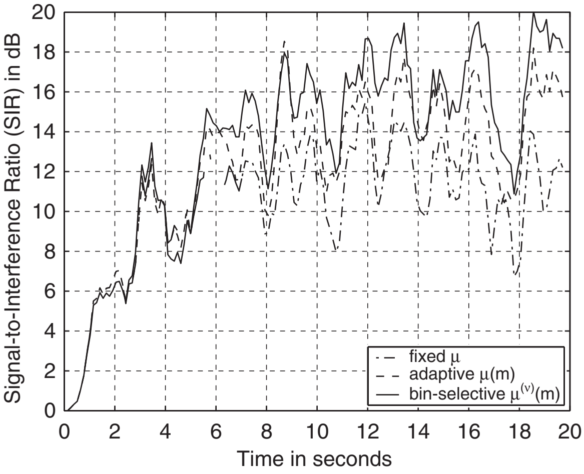

Fig. 8. Evaluation of different stepsize controls ($T_{60} = 50$ ms).

To investigate the effect of the chosen Sylvester constraint and to show the necessity of noncausal filters, we use two different geometric setups as already depicted in Fig. 5. In the first setup the speakers are located at ±45° which implies that no noncausal delay is necessary and thus, both SC_C and SC_R can be applied (see Section 3, Steps 3 and 4). In the second setup, the angle between the sources is only 45° and the two sources are *located in one half plane*. In the enclosure with $T_{60} = 50$ ms the angles were chosen as +45° and +90° whereas for the office room ($T_{60} = 250$ ms) the speakers were located at +15° and +60°. For both enclosures the second scenario requires one noncausal filter w_{12} or w_{21} and thus, an initialization with a shifted unit impulse is required which means that only SC_R can be used. In the experiments a shift of 20 samples is used.

In Figs. 9 and 10 it can be seen that the maximum achievable performance depends on the reverberation time. BSS can be seen as blind MIMO system identification [6] and thus more reverberation increases the number of filter taps to be identified. Moreover, we compare the performance for both Sylvester constraints SC_C and SC_R. For causal scenarios, i.e., source locations in two half planes of the room (Fig. 5a) it can be observed that SC_C is

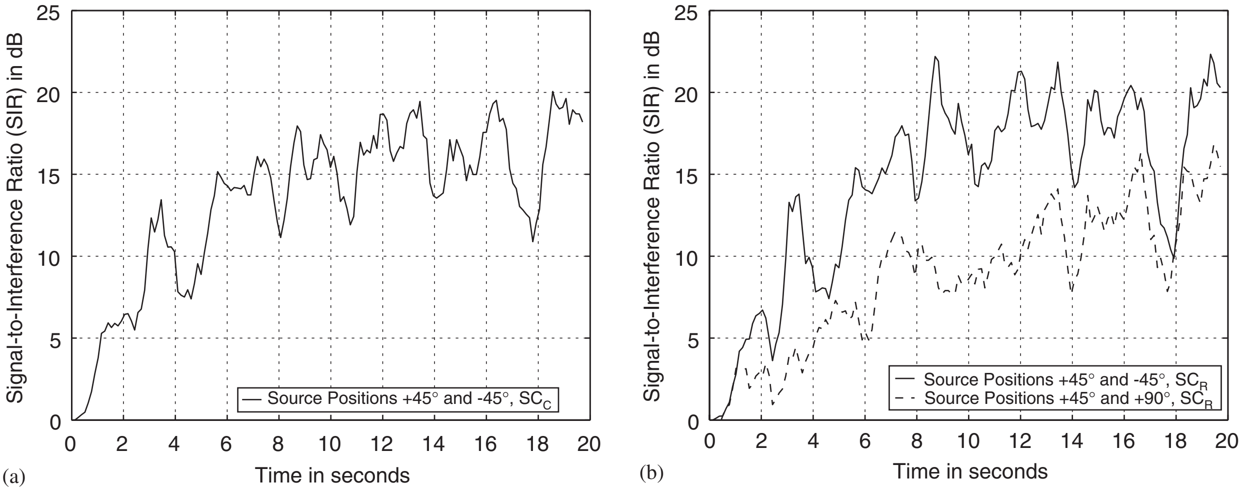

Fig. 9. SC_C versus SC_R for two geometric setups ($T_{60} = 50$ ms): (a) Sylvester constraint SC_C, (b) Sylvester constraint SC_R.

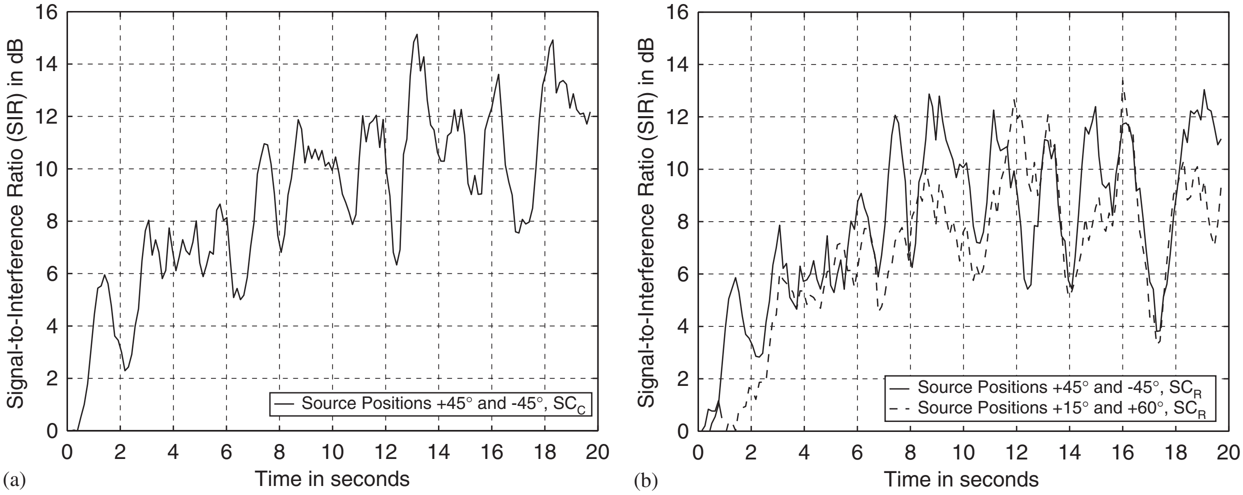

Fig. 10. SC_C versus SC_R for two geometric setups ($T_{60} = 250\,\text{ms}$): (a) Sylvester constraint SC_C, (b) Sylvester constraint SC_R.

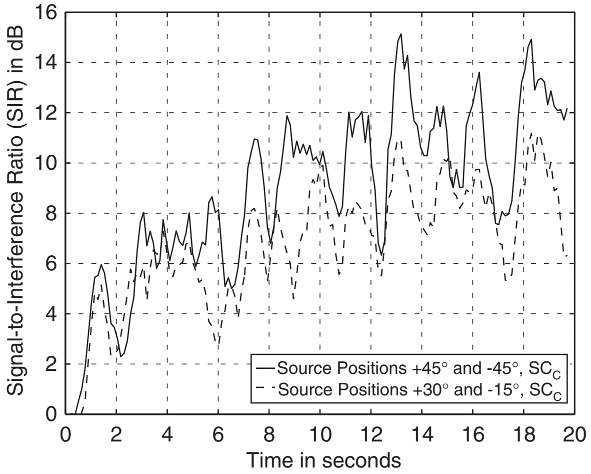

Fig. 11. Effect of varying distance between sources in a causal setup ($T_{60} = 250\,\text{ms}$).

sufficient and gives good results (Figs. 9a and 10a). As noted above, for noncausal setups the algorithm using SC_C cannot converge and thus the SIR would fluctuate around 0 dB. In Figs. 9b and 10b the performance of the algorithm using SC_R is depicted for causal and noncausal setups. Despite the added generality of SC_R there is no significant loss in performance for causal scenarios compared to using SC_C. In addition we can now treat noncausal scenarios, i.e., setups where sources are located in one half plane (Fig. 5b). Note that this case is in general more difficult for two reasons: On the one hand a noncausal filter is needed as discussed above. On the other hand the sources are usually closer together which makes the mixing system more ill-conditioned and in general reduces the maximum achievable SIR.

In Fig. 11 it can be seen that the reduction of maximum achievable performance for smaller distances between the sources is independent of the causality problem and therefore applies also to causal setups. Thus, the separation performance of algorithms using SC_C and SC_R are comparable. The advantage of algorithms based on SC_C is the increased robustness, whereas SC_R-based algorithms are more general due to the possibility of adapting noncausal demixing filters.

7.2. Car environment

The two-element array was mounted at the rear mirror which was directed towards the driver in a Skoda Felicia car. The reverberation time was $T_{60} = 50\,\text{ms}$. The male and female speaker signals were convolved with the measured impulse responses of the car from the driver and co-driver positions, respectively. The angles of both speakers relative to the normal axis of the array were approximately $+25^\circ$ and -65°. Car noise was recorded while driving through a suburban area at a speed of 60 km/h. This noise type exhibits diffuse sound field characteristics, i.e., it is spatially correlated for low frequencies but uncorrelated for higher frequencies. More details on car noise characteristics can be found, e.g., in [32]. Furthermore, spatiotemporally uncorrelated white noise was used. The speech mixtures were additively

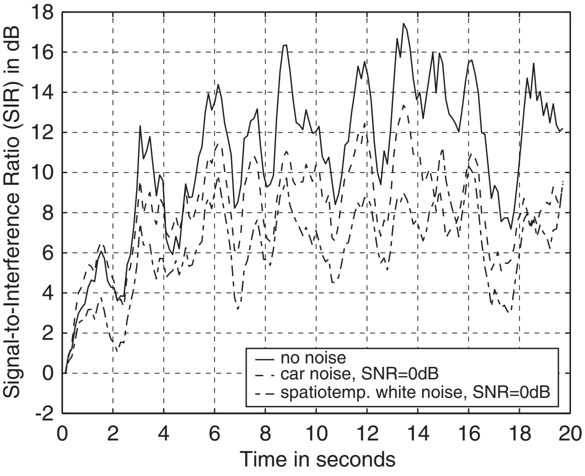

Fig. 12. BSS performance for noisy data (SNR = 0 dB).

mixed with each noise type at an SNR of 0 dB. The parameters of the algorithm have been chosen analogously to Section 7.1. Due to the geometric setup causal filters are adequate. Therefore, we chose the slightly more robust Sylvester constraint SC_C.

In Fig. 12 the influence of *car noise* and *spatiotemporally uncorrelated noise* on the BSS algorithm is shown. Compared to the noiseless case (solid) the separation performance of the noisy cases is reduced. However, especially for the car noise even for a low SNR of 0 dB good separation results are obtained. It should be noted that due to the background noise we can observe masking effects which, from a perceptual point of view, lead to a similar *perceived* suppression of the interfering speaker in the noisy case compared to the noiseless case. The reduced SIR is caused by the bias of the cross-correlation matrices introduced by the noise term (Section 5). It can be seen that diffuse car noise affects the BSS algorithm less than the white noise. The reason is that for a given SNR the bias of the correlation matrices is distributed among all elements of $\mathbf{R}_{\mathbf{yy}}$ in the case of diffuse noise, whereas for spatiotemporally uncorrelated noise mainly the auto-correlation matrices $\mathbf{R}_{\mathbf{y}_p\mathbf{y}_p}$ are affected.

Due to the minimization of the cross-correlation of the outputs also the spatially correlated noise term is partially suppressed. Thus, for diffuse car noise the BSS algorithm achieves—in addition to the SIR improvement—also an average SNR improvement for both channels of 7.3 dB. The SNR for every channel is computed as the ratio of the signal power of the desired speaker to the power of the noise signal. Then, the SNR improvement in dB is calculated as: $10\log_{10}(\mathrm{SNR}_{\mathrm{output}}/\mathrm{SNR}_{\mathrm{input}})$. For spatiotemporally uncorrelated noise, however, no SNR improvement can be observed as the noise signals are uncorrelated between the sensors and thus the BSS criterion cannot achieve any noise reduction.

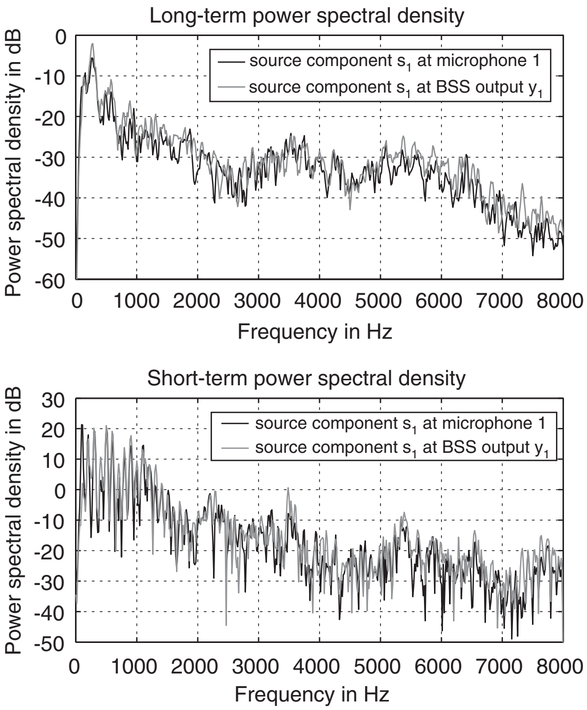

Fig. 13. Comparison of the spectrum of the desired signal s_1 at the first microphone and at the BSS output y_1 using long-term power-spectral densities (psds) averaged over 10 s and short-time psds of one block of length 1024.

7.3. Output signal quality

As pointed out in Section 1 a common problem of traditional MCBD algorithms (e.g., [2,8]) applied to speech signals is the reduced signal quality due to whitening of the desired signal. In contrast to those algorithms the proposed method performs blind MIMO system identification for cancellation of the interfering signal components as shown in [6]. Thus, the whitening effect is avoided which will be examined in this section by exemplarily considering the influence of the demixing system on one individual source. In Fig. 13 the spectrum of the source signal s_1 picked up by the first microphone is compared to the spectrum of the corresponding signal component obtained at the BSS output y_1. As in general BSS is not aiming at dereverberation, it is desirable that the spectrum of the desired source signal at the microphone is not distorted by the BSS processing. The results in Fig. 13 are shown exemplarily for the BSS output signals obtained by the experiment described in Fig. 10a where two sources in the office room ($T_{60} = 250\,\mathrm{ms}$) are located at ±45°. It can be seen in Fig. 13 that for both, using long-term and short-term power spectral densities, the spectra of the desired signal at the microphone and at the BSS output match very well. These results confirm that this broadband BSS algorithm avoids the whitening effect.

8. Conclusions

In this paper, we presented a low-complexity real-time implementation of a BSS algorithm based on a general class of broadband algorithms. Depending on the application, the resulting system allows to adapt causal or noncausal FIR demixing filters, which account for the reverberation of the mixing system. Moreover, a stepsize control is presented which enhances the robustness against reverberation and background noise as was verified by several experiments in real-world scenarios.

Appendix A. Derivation of block-online update rule

A numerical offline optimization for $\mathbf{W}$ is given for any choice of $\beta(i,m)$ by

$$\mathbf{W}^j(m) = \mathbf{W}^{j-1}(m) - \tilde{\mu}\Delta\mathbf{W}^j(m), \quad j = 1,\ldots,j_{\max}. \tag{A.1}$$

From (A.1) a recursive block-online algorithm can be derived by inserting the natural gradient (13) and the block-online weighting function (16) which leads to

$$\begin{aligned}\Delta\mathbf{W}^j(m) &= \nabla^{\mathrm{NG}}_{\mathbf{W}}\mathcal{J}(m,\mathbf{W})^{j-1} & \text{(A.2)}\\ &= 2\sum_{i=0}^{\infty}\frac{1-\lambda}{K}\sum_{m'=0}^{m}\lambda^{m-m'}\varepsilon_{m'K,m'K+K-1}(i)\\ &\quad\cdot\mathcal{Q}(i,\mathbf{W}^{j-1}(m')) & \text{(A.3)}\\ &= 2(1-\lambda)\sum_{m'=0}^{m}\lambda^{m-m'}\frac{1}{K}\\ &\quad\cdot\sum_{i=m'K}^{m'K+K-1}\mathcal{Q}(i,\mathbf{W}^{j-1}(m')). & \text{(A.4)}\end{aligned}$$

The last sum of (A.4) is now denoted as the offline update term

$$\tilde{\mathcal{Q}}(m',\mathbf{W}^{j-1}(m')) = \frac{1}{K}\sum_{i=m'K}^{m'K+K-1}\mathcal{Q}(i,\mathbf{W}^{j-1}(m')), \tag{A.5}$$

which contains K update terms $\mathcal{Q}$ and corresponds to (18). This simultaneous optimization for K blocks allows to exploit the nonstationarity of the source signals as for each block the source statistics change and thus new conditions are generated.

The iterative offline update (A.1) can also be expressed in an explicit manner

$$\mathbf{W}^{j_{\max}}(m) = \mathbf{W}^0(m) - \tilde{\mu}\sum_{j=1}^{j_{\max}}\Delta\mathbf{W}^j(m), \tag{A.6}$$

where $\mathbf{W}^0(m)$ denotes the initialization of the offline algorithm at the mth block. Together with (A.4) and (A.5) the offline update can be written as

$$\begin{aligned}\underbrace{\mathbf{W}^{j_{\max}}(m)}_{=:\frac{\mathbf{W}(m)}{1-\lambda}} &= \mathbf{W}^0(m) - 2\tilde{\mu}(1-\lambda)\sum_{m'=0}^{m}\lambda^{m-m'}\\ &\quad\cdot\sum_{j=1}^{j_{\max}}\tilde{\mathcal{Q}}(m',\mathbf{W}^{j-1}(m')). & \text{(A.7)}\end{aligned}$$

The matrix $\mathbf{W}(m)$ denotes the final demixing matrix containing the FIR filters which are used to separate the sensor signals in the mth block. Additionally a scaling factor $1-\lambda$ is introduced in (A.7) to ensure an unbiased estimation of the demixing matrix $\mathbf{W}(m)$ in the final update rule below in (A.14). To allow for an efficient implementation, we choose the initialization $\mathbf{W}^0(m)$ at the current block as $\mathbf{W}^0(m) = \sum_{m'=0}^{m}\lambda^{m-m'}\mathbf{W}^0(m')$ which leads to

$$\begin{aligned}\mathbf{W}(m) &= (1-\lambda)\sum_{m'=0}^{m}\lambda^{m-m'}\Bigg(\mathbf{W}^0(m') - 2\tilde{\mu}(1-\lambda)\\ &\quad\cdot\sum_{j=1}^{j_{\max}}\tilde{\mathcal{Q}}(m',\mathbf{W}^{j-1}(m'))\Bigg). & \text{(A.8)}\end{aligned}$$

Comparing (A.7) with (A.8) we can see that in (A.7) for every offline iteration $j = 1,\ldots,j_{\max}$, processing of all the data up to the current block m is required. However, by the above chosen initialization it is possible to exchange the order of online and offline part leading to (A.8). In (A.8) the offline iterations denoted by the term in brackets are calculated first and then the online averaging is performed by the outer sum. This reordering allows for an efficient implementation.

The exchange of the order of online and offline part is an approximation which is also used to obtain an efficient recursive algorithm when deriving the RLS algorithm from the Newton–Raphson method [33, p. 329] and [34]. Analogously to [33,34] it is assumed that the minimum of $\nabla_{\mathbf{W}}^{\mathrm{NG}}\mathscr{J}(m-1,\mathbf{W})$ is at $\mathbf{W}(m-1)$ leading to the following approximation:

$$\nabla_{\mathbf{W}}^{\mathrm{NG}}\mathscr{J}(m-1,\mathbf{W}(m-1))=\mathbf{0}. \tag{A.9}$$

Eq. (A.4) can be written recursively and can be approximated using (A.9) yielding

$$\begin{aligned}\Delta\mathbf{W}^{j}(m)&\\ &=\lambda\Delta\mathbf{W}^{j}(m-1)+2(1-\lambda)\tilde{\mathscr{Q}}(m,\mathbf{W}^{j-1}(m)), &\text{(A.10)}\\ &\approx 2(1-\lambda)\tilde{\mathscr{Q}}(m,\mathbf{W}^{j-1}(m)). &\text{(A.11)}\end{aligned}$$

This last expression corresponds to the offline update in (17). Inserting (A.11) into (A.6) leads to the following explicit formulation of an approximated offline update $\tilde{\mathbf{W}}^{j_{\max}}(m)$ which does not exploit the preceding blocks:

$$\tilde{\mathbf{W}}^{j_{\max}}(m)=\mathbf{W}^{0}(m)-2(1-\lambda)\tilde{\mu}\sum_{j=1}^{j_{\max}}\tilde{\mathscr{Q}}(m,\mathbf{W}^{j-1}(m)). \tag{A.12}$$

Now, we see that (A.8) can be expressed using the approximated offline update

$$\begin{aligned}\mathbf{W}(m)&=(1-\lambda)\sum_{m'=0}^{m}\lambda^{m-m'}\tilde{\mathbf{W}}^{j_{\max}}(m') &\text{(A.13)}\\ &=\lambda\mathbf{W}(m-1)+(1-\lambda)\tilde{\mathbf{W}}^{j_{\max}}(m). &\text{(A.14)}\end{aligned}$$

Thus, the final algorithm combines the offline part given by the recursive formulation of (A.12) together with (A.5) and the online part given by (A.14), and is summarized in Section 2.3.

References

[1] A. Hyvaerinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[2] K. Matsuoka, S. Nakashima, Minimal distortion principle for blind source separation, in: International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), San Diego, CA, USA, December 2001.

[3] H. Sawada, R. Mukai, S. Araki, S. Makino, A robust and precise method for solving the permutation problem of frequency-domain blind source separation, IEEE Trans. Speech Audio Process. 12 (5) (September 2004) 530–538.

[4] H. Buchner, R. Aichner, W. Kellermann, A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics, IEEE Trans. Speech Audio Process. 13 (1) (January 2005) 120–134.

[5] H. Buchner, R. Aichner, W. Kellermann, Blind source separation for convolutive mixtures: a unified treatment, in: J. Benesty, Y. Huang (Eds.), Audio Signal Processing for Next-generation Multimedia Communication Systems, Kluwer Academic Publishers, Boston, MA, 2004, pp. 255–293.

[6] H. Buchner, R. Aichner, W. Kellermann, Relation between blind system identification and convolutive blind source separation, in: Proceedings of the Joint Workshop on Hands-Free Communication and Microphone Arrays, Piscataway, NJ, USA, March 2005.

[7] M.I. Gürelli, C.L. Nikias, EVAM: an eigenvector-based algorithm for multichannel blind deconvolution of input colored signals, IEEE Trans. Signal Process. 43 (1) (January 1995) 134–149.

[8] S.C. Douglas, Blind separation of acoustic signals, in: M. Brandstein, D. Ward (Eds.), Microphone Arrays: Signal Processing Techniques and Applications, Springer, Berlin, 2001, pp. 355–380.

[9] F. Asano, S. Ikeda, Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation, in: International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), 2000, pp. 411–415.

[10] R. Mukai, H. Sawada, S. Araki, S. Makino, Robust real-time blind source separation for moving speakers using blockwise ICA and residual crosstalk subtraction, in: Proceedings of the International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), Nara, Japan, April 2003, pp. 975–980.

[11] R. Aichner, H. Buchner, W. Kellermann, On the causality problem in time-domain blind source separation and deconvolution algorithms, in: IEEE International Conference Acoustics, Speech, Signal Processing (ICASSP), vol. 5, Philadelphia, PA, USA, March 2005, pp. 181–184.

[12] A. Cichocki, S.-I. Amari, Adaptive Blind Signal and Image Processing, Wiley, Chichester, 2002.

[13] S. Douglas, A. Cichocki, Adaptive step size techniques for decorrelation and blind source separation, in: Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 1998, pp. 1191–1195.

[14] H. Gish, D. Cochran, Generalized coherence, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 5, April 1988, pp. 2745–2748.

[15] H. Buchner, R. Aichner, W. Kellermann, A generalization of a class of blind source separation algorithms for convolutive mixtures, in: Proceedings of the International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), Nara, Japan, April 2003, pp. 945–950.

[16] K. Matsuoka, M. Ohya, M. Kawamoto, Neural net for blind separation of nonstationary signals, IEEE Trans. Neural Networks 8 (3) (1995) 411–419.

[17] S. Amari, A. Cichocki, H.H. Yang, A new learning algorithm for blind signal separation, in: D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems, vol. 8, MIT Press, Cambridge, 1996, pp. 757–763.

[18] S. Haykin, Adaptive Filter Theory, fourth ed., Prentice-Hall Inc., Englewood Cliffs, NJ, 2002.

[19] E. Moulines, O.A. Amrane, Y. Grenier, The generalized multidelay adaptive filter: structure and convergence analysis, IEEE Trans. Signal Process. 43 (1) (January 1995) 14–28.

[20] J.D. Markel, A.H. Gray, Linear Prediction of Speech, Springer, Berlin, 1976.

[21] R. Aichner, S. Araki, S. Makino, T. Nishikawa, H. Saruwatari, Time-domain blind source separation of non-stationary convolved signals by utilizing geometric beamforming, in: IEEE International Workshop on Neural Networks for Signal Processing (NNSP), Martigny, Switzerland, September 2002, pp. 445–454.

[22] T. Nishikawa, H. Saruwatari, K. Shikano, Comparison of time-domain ICA, frequency-domain ICA and multistage ICA for blind source separation, in: Proceedings of the European Signal Processing Conference, vol. 2, September 2002, pp. 15–18.

[23] A. Cichocki, R. Unbehauen, Neural Networks for Optimization and Signal Processing, Wiley, Chichester, 1994.

[24] H. Buchner, J. Benesty, W. Kellermann, Multichannel frequency-domain adaptive filtering with application to acoustic echo cancellation, in: J. Benesty, Y. Huang (Eds.), Adaptive Signal processing: Application to Real-world Problems, Springer, Berlin, January 2003, pp. 95–128.

[25] B. van Veen, K. Buckley, Beamforming: a versatile approach to spatial filtering, IEEE Trans. Acoust., Speech, Signal Process. (April 1988) 4–24.

[26] H.V. Sorensen, D.L. Jones, M.T. Heideman, C.S. Burrus, Real-valued fast Fourier transform algorithms, IEEE Trans. Acoust., Speech, Signal Process. 35 (6) (June 1987) 849–863.

[27] R. Martin, Noise power spectral density estimation based on optimal smoothing and minimum statistics, IEEE Trans. Speech Audio Process. 9 (5) (July 2001) 504–512.

[28] N.N. Schraudolph, X. Giannakopoulos, Online independent component analysis with local learning rate adaptation, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, 2000, pp. 789–795.

[29] R. Rojas, Neural Networks, Springer, Berlin, 1996.

[30] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Allcon, Accelerating the convergence of the back propagation method, Biol. Cybern. 59 (1988) 257–263.

[31] R. Aichner, H. Buchner, F. Yan, W. Kellermann, Real-time convolutive blind source separation based on a broadband approach, in: International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), Granada, Spain, September 2004, pp. 833–840.

[32] R. Martin, Small microphone arrays with postfilters for noise and acoustic echo reduction, in: M. Brandstein, D. Ward (Eds.), Microphone Arrays: Signal Processing Techniques and Applications, Springer, Berlin, 2001, pp. 255–279.

[33] H.V. Söderström, P. Stoica, System identification, Prentice-Hall International Series in Systems and Control Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[34] H. Buchner, J. Benesty, T. Gänsler, W. Kellermann, Robust extended multidelay filter and double-talk detector for acoustic echo cancellation, IEEE Trans. Speech Audio Process. 13 (2006) in press.