REAL-TIME BLIND SOURCE SEPARATION FOR MOVING SPEAKERS USING BLOCKWISE ICA AND RESIDUAL CROSSTALK SUBTRACTION

Ryo Mukai *Hiroshi Sawada* *Shoko Araki* *Shoji Makino*

NTT Communication Science Laboratories, NTT Corporation
2–4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619–0237, Japan
{ryo,sawada,shoko,maki}@cslab.kecl.ntt.co.jp

ABSTRACT

This paper describes a real-time blind source separation (BSS) method for moving speech signals in a room. Our method employs frequency domain independent component analysis (ICA) using a blockwise batch algorithm in the first stage, and the separated signals are refined by postprocessing using crosstalk component estimation and non-stationary spectral subtraction in the second stage. The blockwise batch algorithm achieves better performance than an online algorithm when sources are fixed, and the postprocessing compensates for performance degradation caused by source movement. Experimental results using speech signals recorded in a real room show that the proposed method realizes robust real-time separation for moving sources. Our method is implemented on a standard PC and works in realtime.

1. INTRODUCTION

Blind source separation (BSS) is a technique for estimating original source signals using only observed mixtures. The BSS of audio signals has a wide range of applications including noise robust speech recognition, hands-free telecommunication systems and high-quality hearing aids. In most realistic applications, the source signal location may change, and the mixing system is time-varying. Although a large number of studies have been undertaken on BSS based on ICA [1, 2, 3], only few studies have been made on BSS for moving source signals [4, 5, 6, 7]. Indeed an online algorithm can track a time-varying system, however, in general, its performance is worse than a batch algorithm when the system becomes stationary. Although we are dealing with moving sources, we do not want to degrade the performance for fixed sources.

In this paper, we propose a robust real-time BSS method that employs frequency domain ICA using a blockwise batch algorithm in the first stage, and the postprocessing of crosstalk component estimation and non-stationary spectral subtraction in the second stage. When we adopt a blockwise frequency domain ICA, we need to solve a permutation problem for every block, and this is a time consuming process especially when the block length is short. We use an algorithm based on analytical calculation of null directions to solve the permutation problem quickly. Another problem inherent to batch algorithms is an input-output delay. To reduce the delay, we use a technique for computing output signal without waiting for the calculation of the separating system to be completed. These techniques are useful for realizing low-delay real-time BSS.

The blockwise batch algorithm achieves better separation performance than an online algorithm for fixed source signals, but the performance declines for moving sources. As we pointed out in [8], the solution of ICA works like an adaptive beamformer, which forms a spatial null towards a jammer signal. This characteristic means that BSS using ICA is fragile as regards a moving jammer signal but robust with respect to a moving target signal. Utilizing this nature, we can estimate residual crosstalk components even when a jammer signal moves. To compensate for the degradation when a jammer signal moves, we employ postprocessing in the second stage.

Experimental results using speech signals recorded in a room show the effectiveness of the method in realizing robust real-time separation.

2. ICA BASED BSS OF CONVOLUTIVE MIXTURES

In this section, we briefly review the BSS algorithm that uses frequency domain ICA and formulate a blockwise batch algorithm including an online algorithm as a special case. We also describe a fast algorithm for solving permutation problems, which is necessary for real-time processing.

2.1. Frequency domain ICA

When the source signals are $s_i(t)(i = 1, ..., N)$, the signals observed by microphone j are $x_j(t)(j = 1, ..., M)$, and the separated signals are $y_k(t)(k = 1, ..., N)$, the BSS model can be described by the following equations:

$$x_j(t) = \sum_{i=1}^{N} (\boldsymbol{h}_{ji} * s_i)(t) \tag{1}$$

$$y_k(t) = \sum_{j=1}^{M} (\boldsymbol{w}_{kj} * x_j)(t) \tag{2}$$

where $\boldsymbol{h}_{ji}$ is the impulse response from source i to microphone j, $\boldsymbol{w}_{kj}$ is the coefficient when we assume that a separating system is used as an FIR filter, and $*$ denotes the convolution operator.

A convolutive mixture in the time domain corresponds to instantaneous mixtures in the frequency domain. Therefore, we can apply an ordinary ICA algorithm in the frequency domain to solve a BSS problem in a reverberant environment. Using a short-time discrete Fourier transform (STDFT) for (1), the model is approximated as:

$$\boldsymbol{X}(\omega, n) = \boldsymbol{H}(\omega)\boldsymbol{S}(\omega, n), \tag{3}$$

where, ω is the angular frequency, and n represents the frame index. The separating process can be formulated in each frequency bin as:

$$\boldsymbol{Y}(\omega, n) = \boldsymbol{W}(\omega)\boldsymbol{X}(\omega, n), \tag{4}$$

where $\boldsymbol{S}(\omega, n) = [S_1(\omega, n), ..., S_N(\omega, n)]^T$ is the source signal in frequency bin ω, $\boldsymbol{X}(\omega, n) = [X_1(\omega, n), ..., X_M(\omega, n)]^T$ denotes the observed signals, $\boldsymbol{Y}(\omega, n) = [Y_1(\omega, n), ..., Y_N(\omega, n)]^T$ is the estimated source signal, and $\boldsymbol{W}(\omega)$ represents the separating matrix. $\boldsymbol{W}(\omega)$ is determined so that $Y_i(\omega, n)$ and $Y_j(\omega, n)$ become mutually independent.

To calculate the separating matrix $\boldsymbol{W}$, we use an optimization algorithm based on the minimization of the mutual information of $\boldsymbol{Y}$. The optimal $\boldsymbol{W}$ is obtained by the following iterative equation using the natural gradient approach [9] :

$$\boldsymbol{W}^{(i+1)} = \boldsymbol{W}^{(i)} + \mu[\boldsymbol{I} - \langle \Phi(\boldsymbol{Y})\boldsymbol{Y}^H \rangle]\boldsymbol{W}^{(i)}, \tag{5}$$

where i is an index for the iteration, $\boldsymbol{I}$ is an identity matrix, μ is a step size parameter, $\langle\cdot\rangle$ denotes the averaging operator, and $\Phi(\cdot)$ is a nonlinear function. Because the signals have complex values in the frequency domain, we use a polar coordinate based nonlinear function, which is effective for fast convergence especially when the number of input data samples is small [10]:

$$\Phi(\boldsymbol{Y}) = \tanh(g \cdot \mathrm{abs}(\boldsymbol{Y}))e^{\jmath \arg(\boldsymbol{Y})}, \tag{6}$$

where g is a gain parameter that controls the nonlinearity.

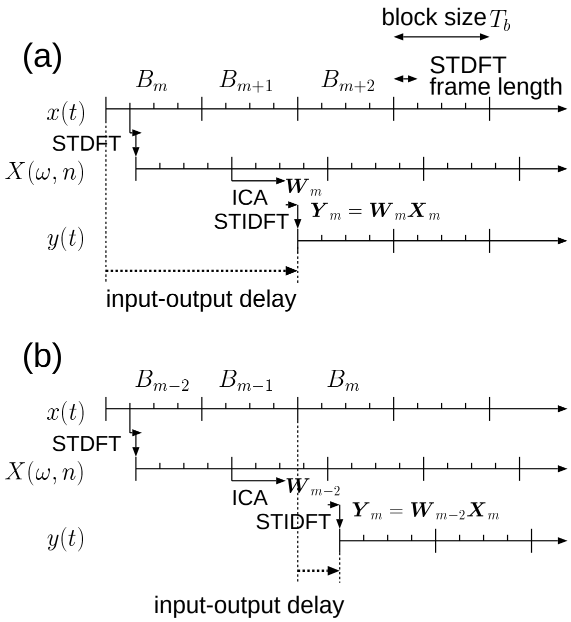

Fig. 1. Input-output delay of (a) BSS using ordinary blockwise batch algorithm, and (b) BSS without waiting for calculation of $\boldsymbol{W}_m$.

2.2. Blockwise batch algorithm

In order to track the time-varying mixing system, we update the separating matrix for each time block $B_m = \{t : (m-1)T_b \le t < mT_b\}$, where T_b is the block size, and m represents the block index $(m \ge 1)$.

Koutras et al. have proposed a similar method in the time domain [5]. When T_b equals the STDFT frame length, this procedure can be considered an online algorithm in the frequency domain.

We use the separating matrix of the previous block as the initial iteration value for a new block, *i.e.*, $\boldsymbol{W}_{m+1}^{(0)}(\omega) = \boldsymbol{W}_m^{(N_I)}(\omega)$, where N_I is the number of iterations for (5). We use a set of two null beamformers as the initial matrix $\boldsymbol{W}_1^{(0)}(\omega)$ for the first block.

The batch algorithm has an inherent delay, because the calculation of $\boldsymbol{W}$ needs to wait for the arrival of a data block. Moreover, the calculation itself also takes time (Fig. 1(a)). However, when the calculation is completed within T_b and we use $\boldsymbol{W}_{m-2}$ for separation of the signals in B_m, we can avoid the delay for waiting and calculation (Fig. 1(b)). This technique can reduce the input-output delay and is suitable for low-delay real-time applications. It seems that this method fails when a source signal moves, but it is actually robust for the moving target signal, which is shown in Sec. 4.3.

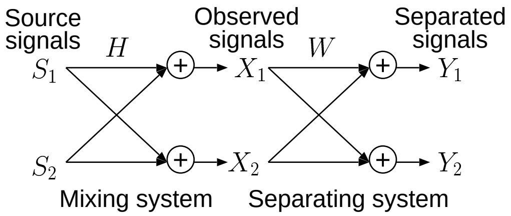

Fig. 2. Model of BSS system ($N = M = 2$).

2.3. Scaling and permutation

Once we have completed the ICA for all frequencies, we need to solve the permutation and scaling problems. Since we are handling signals with complex values, the scaling factors are also complex values. Thus the scaling can be divided into phase scaling and amplitude scaling.

We use a directivity pattern based method to solve the permutation and phase scaling problems. When we consider a separating system as a microphone array, we can write directivity patterns for every frequency bin. The permutation problem is solved so that the null directions are aligned. We can estimate the directions of the source signals from the aligned directivity patterns, and the phase scaling problem is solved so that the phase response of the estimated source direction becomes zero.

In the following sections, we consider a two-input, two-output convolutive BSS problem, *i.e.*, $N = M = 2$ (Fig. 2). When $M = 2$ and the distance between the microphones is sufficiently small to avoid spatial aliasing, the null directions $\theta_i(\omega)$ can be calculated analytically as:

$$\theta_i(\omega) = \arcsin\left(\arg\left(\frac{w_{i1}(\omega)}{w_{i2}(\omega)}\right)\frac{c}{d\cdot\omega}\right), \qquad (7)$$

where $[w_{i1}(\omega)w_{i2}(\omega)]$ is an i-th row vector of $\boldsymbol{W}(\omega)$, d is the distance between microphones and c is the velocity of sound [11]. This method does not require the directivity pattern to be scanned, thus we can solve the permutation problem quickly.

The amplitude scaling problem is solved by using a slightly modified version of the method described in [12]. We calculate the inverse of the separating matrices $\boldsymbol{W}(\omega)^{-1}$, and decide the scaling factors so that the norms of each column of $\boldsymbol{W}(\omega)^{-1}$ become uniform.

3. POSTPROCESSING FOR REFINING SEPARATED SIGNALS

In this section, we briefly summarize the procedure for estimating and subtracting the residual crosstalk component. The algorithm is described in detail in [13]. Figure 3 shows a block diagram of the algorithm.

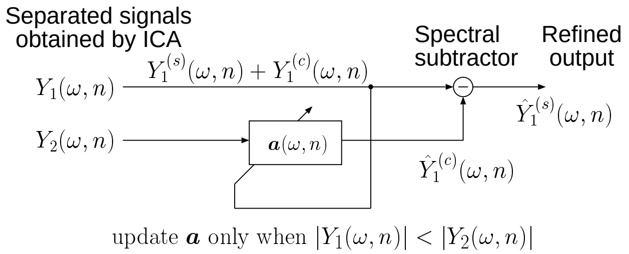
Separated signals obtained by ICA
Spectral subtractor
Refined output
$Y_1(\omega,n)$
$Y_1^{(s)}(\omega,n)+Y_1^{(c)}(\omega,n)$
$Y_2(\omega,n)$
$\boldsymbol{a}(\omega,n)$
$\hat{Y}_1^{(s)}(\omega,n)$
$\hat{Y}_1^{(c)}(\omega,n)$
update $\boldsymbol{a}$ only when $|Y_1(\omega,n)|<|Y_2(\omega,n)|$

Fig. 3. Postprocessing for removing crosstalk component $Y_1^{(c)}$ from Y_1.

We consider that S_1 is a target signal and S_2 is a jammer signal. The separated signal Y_1 consists of a straight component $Y_1^{(s)}$ derived from S_1 and a crosstalk component $Y_1^{(c)}$ derived from S_2. If $Y_1^{(c)}$ is removed from Y_1, the separation performance improves.

We introduce FIR filters $\boldsymbol{a}(\omega,n) = [a_0(\omega,n),..., a_{L-1}(\omega,n)]$ in each frequency bin, where L is the length of the filter. We assume that the power of $Y_1^{(c)}(\omega,n)$ can be approximated as the output of the filter whose input is $Y_2(\omega,n)$. This is formulated as follows:

$$|\hat{Y}_1^{(c)}(\omega,n)|^2 \approx \sum_{k=0}^{L-1} a_k(\omega,n)|Y_2(\omega,n-k)|^2 \qquad (8)$$

The filters are updated by the following selectively normalized LMS algorithm.

$$\Delta\boldsymbol{a}(\omega,n+1) = \qquad (9)$$
$$\begin{cases} \dfrac{\eta}{\delta+||\boldsymbol{u}(\omega,n)||^2}e(\omega,n)\boldsymbol{u}(\omega,n) \\ \quad (\text{if } |Y_1(w,n)| < |Y_2(w,n)|) \\ 0 \text{ (otherwise)} \end{cases}$$

where $\boldsymbol{u}(\omega,n) = [|Y_2(\omega,n)|^2, |Y_2(\omega,n-1)|^2,..., |Y_2(\omega,n-L+1)|^2]^T$ is an input vector and $e(\omega,n) = |Y_1(\omega,n)|^2 - \boldsymbol{a}^T(\omega,n)\boldsymbol{u}(\omega,n)$ is an estimation error. Here, η is a step size parameter and δ is a positive constant to avoid numerical instability when $\boldsymbol{u}$ is very small.

We estimate the power of the residual crosstalk component using (8) and (9), and finally, we obtain an estimation of the straight component as $\hat{Y}_1^{(s)}$ by the following spectral subtraction procedure:

$$\hat{Y}_1^{(s)}(\omega,n) = \qquad (10)$$
$$\begin{cases} (|Y_1(\omega,n)|^2 - |\hat{Y}_1^{(c)}(\omega,n)|^2)^{1/2}\dfrac{Y_1(\omega,n)}{|Y_1(\omega,n)|} \\ \qquad (\text{if } |Y_1(\omega,n)|^2 > |\hat{Y}_1^{(c)}(\omega,n)|^2) \\ 0 \qquad (\text{otherwise}) \end{cases}$$

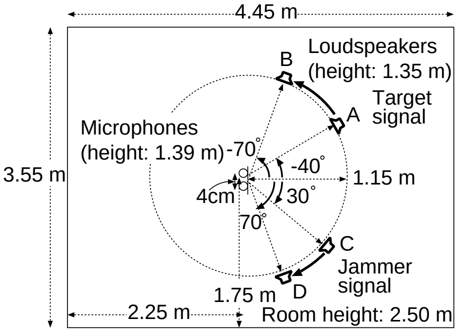

Fig. 4. Layout of room used in experiments.

4. EXPERIMENTS

4.1. Experimental conditions

To examine the effectiveness of the proposed method, we carried out experiments using speech signals recorded in a room. The reverberation time of the room was 130 ms. We used two omni-directional microphones with an inter-element spacing of 4 cm. The layout of the room is shown in Fig 4. The target source signal was first located at A, and then moved to B at a speed of 30 deg/s. The jammer signal was located at C and moved to D at a speed of 40 deg/s.

The step size parameter μ in (5) affects the separation performance of BSS when the block size changes. We chose μ to optimize the performance for each block size. Other conditions are summarized in Table 1.

We assumed the straight component $y_1^{(s)}$ as a signal, and the difference between the output signal and the straight component as interference. We defined the output signal-to-interference ratio (SIR$_O$) in the time domain as follows:

$$\mathrm{SIR}_O \equiv 10\log\frac{\sum_t |y_1^{(s)}(t)|^2}{\sum_t |y_1(t) - y_1^{(s)}(t)|^2}\ (\mathrm{dB}). \qquad (11)$$

Similarly, the input SIR (SIR$_I$) is defined as,

$$\mathrm{SIR}_I \equiv 10\log\frac{\sum_t \sum_{i=1}^{2} |(\boldsymbol{h}_{i1} * s_1)(t)|^2}{\sum_t \sum_{i=1}^{2} |(\boldsymbol{h}_{i2} * s_2)(t)|^2}\ (\mathrm{dB}). \qquad (12)$$

We use SIR $=$ SIR$_O$ $-$ SIR$_I$ as a performance measure. This measurement is consistent with the performance evaluation of BSS in which the crosstalk component is assumed as interference. We measured SIRs with 30 combinations of source signals using three male and three female speakers, and averaged them.

4.2. Performance for fixed sources

Although we are dealing with moving sources, we do not want the performance for fixed sources to deteriorate. First, we measured the BSS performance using ICA without post-processing. Figure 5 shows the average and standard deviation of SIR for fixed sources (the target is at A and the jammer at C in Fig. 4). This indicates that the blockwise batch algorithm outperforms the online algorithm (in which μ is tuned to optimize the performance), when we use the update equation (5). In addition, the deviation of the batch algorithm is smaller than that of the online algorithm. This is why we adopt the blockwise batch algorithm in the first stage. We used T_b = 1.0 sec. in the following experiments.

Table 1. Experimental conditions

Common	Sampling rate = 8 kHz Window = hanning Reverberation time T_R=130 ms
ICA part	Frame length T_{ICA} = 1024 point (128 ms) Frame shift = 256 point (32ms) g = 100.0 Number of iterations N_I = 100 Block size T_b = 1 s
Post processing part	Frame length T_{SS} = 1024 point (128 ms) Frame shift = 64 point (8 ms) $\eta = 0.1, \delta = 0.01$

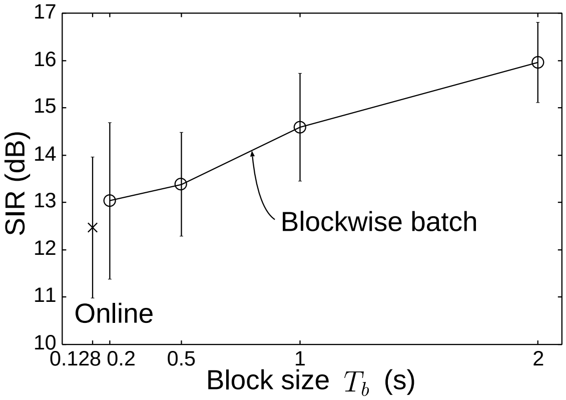

Fig. 5. Average and standard deviation of SIR for fixed sources

4.3. Moving target and moving jammer

Before considering the result obtained with the postprocessing method, we investigate the BSS performance for moving sources using the blockwise batch algorithm. Figure 6 shows the SIR for a moving target (solid line) and for a moving jammer (dotted line). We can see that the SIR is not degraded even when the target moves. By contrast, jammer movement causes a decline in the SIR.

This can be explained by the directivity pattern of the separating system obtained by ICA. The solution of fre-

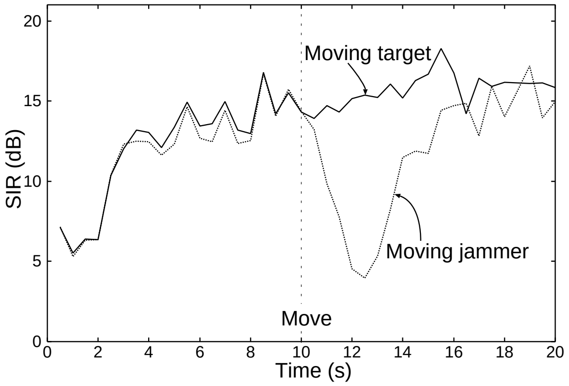

Fig. 6. SIR of blockwise batch algorithm without postprocessing. Target and jammer signals moved at 10 sec.

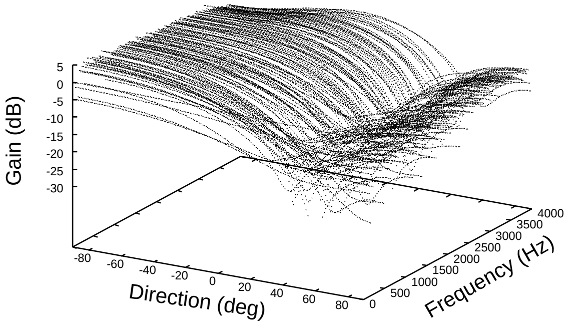

Fig. 7. Directivity pattern of separating system obtained by frequency domain ICA

quency domain BSS works in the same way as an adaptive beamformer, which forms a spatial null towards a jammer signal (Fig. 7). Because of this characteristic, BSS using ICA is robust as regards a moving target signal but fragile with respect to a moving jammer signal.

4.4. Performance of blockwise batch algorithm with postprocessing

The most important factor when estimating the crosstalk component $Y_1^{(c)}$ using (8) and (9) is Y_2, and Y_2 is estimated robustly even when S_2 moves, because S_2 is a target signal for Y_2. Therefore, postprocessing works robustly even when the jammer signal S_2 moves.

Figure 8 shows the SIR of blockwise batch algorithm with postprocessing when the jammer signal moves (solid line). We can see that the SIR is improved by the postprocessing, and the drop of the SIR when the jammer moves is reduced. This result shows that our postprocessing method can compensate the fragility of the blockwise batch algorithm when a jammer signal moves. Although crosstalk components still remaining in the postprocessed output signal sometimes make a musical noise, the power is much smaller than ordinary spectral subtraction.

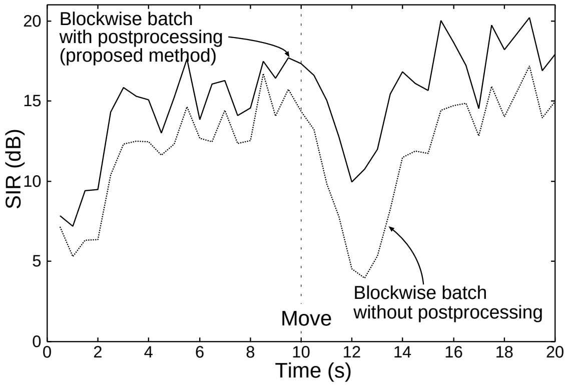

Fig. 8. Effect of postprocessing. Jammer signal moved from C to D at 10 sec.

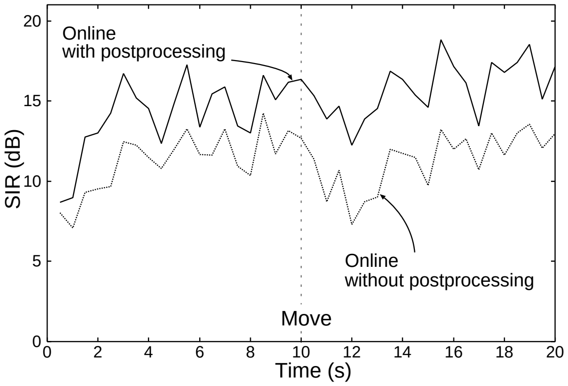

Fig. 9. Performance of online algorithm with and without postprocessing. Jammer signal moved from C to D at 10 sec.

4.5. Performance of online algorithm

Figure 9 shows the SIR of online algorithm with and without postprocessing. The online algorithm is more stable than blockwise algorithm, however the performance is worse when the sources are stationary, as we described in Sec. 4.2. The postprocessing is also effective for this case, thus we may choose the algorithm in the first stage according to requirements of the application.

5. CONCLUSION

We proposed a robust real-time BSS method for moving source signals. The combination of the blockwise batch and the postprocessing realizes a robust low-delay real-time BSS. We can solve a permutation problem quickly by using analytical calculation of null directions, and this technique is useful for solving convolutive BSS problems in realtime. Postprocessing using crosstalk component estimation and non-stationary spectral subtraction improves the separation performance and reduces the performance deterioration when a jammer signal moves. Experimental results using speech signals recorded in a room showed the effectiveness of the proposed method.

ACKNOWLEDGEMENT

We thank Dr. Shigeru Katagiri for his continuous encouragement.

6. REFERENCES

[1] A. J. Bell and T. J. Sejnowski, “An information-maximization approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[2] S. Haykin, Ed., *Unsupervised Adaptive Filtering*, John Wiley & Sons, 2000.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[4] J. Anemüller and T. Gramss, “On-line blind separation of moving sound sources,” in *Proc. of Intl. Conf. on Independent Component Analysis and Blind Source Separation (ICA’99)*, 1999, pp. 331–334.

[5] A. Koutras, E. Dermatas, and G. Kokkinakis, “Blind speech separation of moving speakers in real reverberant environment,” in *Proc. of ICASSP’00*, 2000, pp. 1133–1136.

[6] I. Kopriva, Z. Devcic, and H. Szu, “An adaptive short-time frequency domain algorithm for blind separation of non-stationary convolved mixtures,” in *Proc. of IJCNN’01*, 2001, pp. 424–429.

[7] K. E. Hild II, D. Erdogmus, and J. C. Principe, “Blind source separation of time-varying, instantaneous mixtures using an on-line algorithm,” in *Proc. of ICASSP’02*, 2002, pp. 993–996.

[8] S. Araki, S. Makino, R. Mukai, and H. Saruwatari, “Equivalence between frequency domain blind source separation and frequency domain adaptive null beamformers,” in *Proc. of Eurospeech’01*, 2001, pp. 2595–2598.

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

[10] H. Sawada, R. Mukai, S. Araki, and S. Makino, “Polar coordinate based nonlinear function for frequency-domain blind source separation,” in *Proc. of ICASSP’02*, 2002, pp. 1001–1004.

[11] H. Sawada, R. Mukai, S. Araki, and S. Makino, “A robust approach to the permutation problem of frequency-domain blind source separation,” in *Proc. of ICASSP’03*, 2003, submitted.

[12] F. Asano and S. Ikeda, “Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation,” in *Proc. of Intl. Workshop on Independent Component Analysis and Blind Signal Separation (ICA’00)*, 2000, pp. 411–415.

[13] R. Mukai, S. Araki, H. Sawada, and S. Makino, “Removal of residual crosstalk components in blind source separation using LMS filters,” in *Proc. of NNSP’02*, 2002, pp. 435–444.