A Near Real-Time Approach for Convolutive Blind Source Separation

Shuxue Ding, *Member, IEEE*, Jie Huang, *Member, IEEE*, Daming Wei, *Member, IEEE*, and Andrzej Cichocki, *Member, IEEE*

***Abstract*—In this paper, we propose an algorithm for real-time signal processing of convolutive blind source separation (CBSS), which is a promising technique for acoustic source separation in a realistic environment, e.g., room/office or vehicle. First, we apply an overlap-and-save (sliding windows with overlapping) strategy that is most suitable for real-time CBSS processing; this approach can also aid in solving the permutation problem. Second, we consider the issue of separating sources in the frequency domain. We introduce a modified correlation matrix of observed signals and perform CBSS by diagonalization of the matrix. Third, we propose a method that can diagonalize the modified correlation matrix by solving a so-called normal equation for CBSS. One desirable feature of our proposed algorithm is that it can solve the CBSS problem *explicitly*, rather than stochastically, as is done with conventional algorithms. Moreover, a real-time separation of the convolutive mixtures of sources can be performed. We designed several simulations to compare the effectiveness of our algorithm with its counterpart, the gradient-based approach. Our proposed algorithm displayed superior convergence rates relative to the gradient-based approach. We also designed an experiment for testing the efficacy of the algorithm in real-time CBSS processing aimed at separating acoustic sources in realistic environments. Within this experimental context, the convergence time of our algorithms was substantially faster than that of the gradient-based algorithms. Moreover, our algorithm converges to a much lower value of the cost function than that of the gradient-based algorithm, ensuring better performance.**

***Index Terms*—Convolutive blind source separation (CBSS), explicit algorithm, modified correlation matrix, real-time processing.**

I. Introduction

CONVOLUTIVE blind source separation (CBSS) of acoustic signals in a realistic environment is a rather challenging problem [1], since there are many unknown reflective paths from each audio source to each sensor, and sources are mixed in a complicated form. The typical approach to realize source separation is, first, to model a real-world superposition of audio sources using a mixture of delayed and filtered source signals. Such filters vary from several hundreds to several thousands of taps to faithfully model a typical environment, e.g., a typical room or office. To successfully separate sources, one needs to estimate as accurately as possible the relative delays between paths and the attenuations of all paths. The challenge is in accurately learning optimal values in the complicated space for these delays and attenuations.

Recently, some encouraging results in acoustic BSS have been reported [1]–[18]. However, most of the algorithms that have been proposed work well only in batch processing, even though real-time processing is required by most applications.

In contrast to batch processing, three aspects in particular must be considered in real-time processing. (a) The processing must be fast enough to be finished within the elapsed time of the processed signal. Therefore, very complicated calculations must be avoided during the processing. (b) At each time instant of processing, only a limited number of samples can be used for estimating the separation parameters. In other words, the key problem here is how to estimate the separation parameters accurately with only a few samples. (c) Since most realistic mixing environments can vary within the time that CBSS is being processed, the processing must converge fast enough to ensure tracking ability. A reasonably useful real-time CBSS algorithm should satisfy these three fundamental conditions.

The conventional approach for real-time CBSS is to adapt the filter weights by a gradient search or by another kind of stochastic search in order to minimize a suitable cost function. Although this approach has been proposed for batch processing [1], [11], [17], it can also be applied to real-time processing [13]–[15] in some special cases. Indeed, it works with moderate success for separations of independent identically distributed (I.I.D.) sources. In this case, the conventional approach satisfies the three conditions for real-time CBSS. However, it does not work so well when applied to signals with temporal structures, e.g., acoustic signals in realistic environments, since the signals are neither I.I.D. nor stationary [1], [14]. The gradient-based approach fails to satisfy both the second and third conditions of real-time CBSS for acoustic signals. Gradient-based algorithms cannot converge sufficiently fast for most realistic applications, despite the fact that the so-called natural gradient approach has greatly improved the convergence of the standard gradient approach [19].

In this paper, we propose an algorithm for real-time acoustic signal processing of CBSS that solves these problems. We consider the problem of separating sources in the frequency domain. Next, we devise an overlap-and-save (sliding windows with overlapping) strategy that is most suitable for real-time

Manuscript received April 22, 2004; revised September 18, 2004 and April 2, 2005. This work was supported in part by Project 16500134, 2004 Grants-In-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science and Technology, Japan. This paper was recommended by Associate Editor V. E. DeBrunner.

S. Ding is with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan, and also with the Brain Science Institute, RIKEN, Saitama 351-0198, Japan (e-mail: sding@u-aizu.ac.jp).

J. Huang and D. Wei are with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City, Fukushima 965-8580, Japan.

A. Cichocki is with the Brain Science Institute, RIKEN, Saitama 351-0198, Japan, and also with the Warsaw University of Technology, Warsaw, Poland.

Digital Object Identifier 10.1109/TCSI.2005.854295

CBSS processing. As it is well known, the BSS in each frequency bin will give estimated sources with arbitrary permutation, which cause a so-called permutation problem when we attempt to reconstruct the sources in the time domain [1]–[5]. By choosing proper parameters, we can also use the overlap-and-save approach to help solve this permutation problem. For the separation in the frequency domain, we introduce a modified correlation matrix of observed signals and perform a BSS by diagonalization of the matrix. This diagonalization can be performed by solving a so-called normal equation for CBSS. One desirable feature of the proposed algorithm is that it can *explicitly* estimate the separating matrix. Using this algorithm, we can perform a near real-time separation of the convolutive mixtures of acoustic sources.

In simulation and experimental studies, we compare the effectiveness of our algorithm and its counterpart, the gradient-based approach. We apply the two algorithms to the same cost function and evaluate the convergence rates of the two algorithms. Our proposed algorithm exhibits superior convergence rates compared to those of the gradient-based approach. We also designed an experiment for applying the algorithm to real-time CBSS processing to separate acoustic sources in realistic environments. This experiment demonstrated that our algorithm converges to a stable separating state within several seconds, which is much faster than the convergence time of the gradient-based algorithm. Moreover, our algorithm converges to a much lower cost function value than that resulting from the gradient-based algorithm.

This paper is organized as follows. In Section II, we formulate the posed problem. We give the mixing and separating model on which we shall work, introduce the modified correlation matrix and its diagonalization, and derive the normal equation for CBSS. In Section III, we consider how to estimate the correlation matrix recursively and how to solve the normal equation explicitly. There are many ways to recursively estimate the correlation matrix. As an example, we present a method that exponentially time-weights the samples in the estimation. In Section IV, we present the simulation results and the experimental results. In Sections V and VI, we discuss and draw our conclusions.

The following notations are used in this paper. A variable is represented by the same lowercase letter in both time domain and in the frequency domain. They are distinguished by their arguments, a sample number in the time domain described by a Latin letter, e.g., n or k, or a frequency bin ω_b for a positive integer b in the frequency domain. Matrices in the time domain are represented by boldface uppercase letters. Two types of vectors exist: (a) original vectors and (b) vectors defined by a row or a column component of a matrix. The former vector is represented by boldface lowercase letter, and the latter vector is represented by the corresponding matrix with a subscript. Re and Im denote the real part and imaginary part of a complex variable, respectively. The superscripts *, ^{T}, and ^{H} denote complex conjugation, transposition, and Hermitian transposition, respectively. $\boldsymbol{I}$ is an identity matrix.

II. Problem Formulation, Criterion, and Method

In this section, we formulate the problem of CBSS and present the proposed criterion and method for the problem. Next, we discuss the logic behind our proposed criterion and method.

A. Convolutive Mixtures and Near Real-Time BSS Problem

We assume the existence of M statistically independent sources $\boldsymbol{s}(t) = [s_1(t), \ldots, s_M(t)]^T$ that have a temporal structure, where t is the time instant. These sources are convolved and mixed in a linear medium, leading to N sensor signals $\boldsymbol{x}(t) = [x_1(t), \ldots, x_N(t)]^T$

$$\boldsymbol{x}(t) = (\boldsymbol{A} * \boldsymbol{s})(t) \tag{1}$$

where the asterisk $*$ denotes the convolution operator and $\boldsymbol{A}$ is an $N \times M$ matrix of filters that describe transmitting channels. For simplicity, we have ignored the additive sensor noise and assume that $N = M$.

The purpose of CBSS is to find a matrix $\boldsymbol{W}$ such that

$$\boldsymbol{y}(t) = (\boldsymbol{W} * \boldsymbol{x})(t) \tag{2}$$

recovers the sources $\boldsymbol{s}(t)$. Here $\boldsymbol{y}(t) = [y_1(t), \ldots, y_N(t)]^T$ is the output vector of the CBSS processing, and $\boldsymbol{W}$ is an $N \times N$ matrix of FIR filters with lengths Q.

In real-time implementations of the CBSS approach, the observation signal vector $\boldsymbol{x}(t)$ is input with the unit of sample block, or in other words, with the unit of sample frame. We use T to denote the size of the block. The shifting length between two neighboring blocks is L, which is not necessarily equal to T. A convenient value for L is to choose $L = Q$. In this case, the overlap between two neighboring blocks is of length $K \equiv T - L$. At each block (although we estimate separation parameters based on the sample block with the length T, as the separation result), we output only the first L samples as one output block. In other words, the block length of the output differs from the block length of the input. Connecting the succession of the output blocks forms the output signal flows.

To the number n input block, we can transform (1) and (2) into the frequency domain,[1]

$$\boldsymbol{x}(\omega_b, n) = \boldsymbol{A}(\omega_b)\boldsymbol{s}(\omega_b, n) \tag{3}$$
$$\boldsymbol{y}(\omega_b, n) = \boldsymbol{W}(\omega_b)\boldsymbol{x}(\omega_b, n) \tag{4}$$

where $\boldsymbol{u}(\omega_b, n) = \mathrm{DTFT}([\boldsymbol{u}(L(n-1)+1), \ldots, \boldsymbol{u}(L(n-1)+T)])(\omega_b, n)$, for a signal vector $u(n)$, and $D_{ij}(\omega_b) = \mathrm{DTFT}(D_{ij})$, for a matrix $\boldsymbol{D} = \{D_{ij}\}$, in each frequency bin $\omega_b = ((b-1)/T)2\pi$, for $b = 1, 2, \ldots, T$. Here, n is a variable position of observable sample frames with length of T. In this paper, DTFT denotes the discrete-time Fourier transform. IDTFT will be used to denote the inverse discrete-time Fourier Transform. In this paper, we choose $T \geq L$, for the reasons explained in the next subsection.

Based on (3) and (4), the mixing becomes instantaneous in each frequency bin.

In real-time implementations of the CBSS approach, it is desirable to start the computation with a suitable initial condition.

[1]Here, similar to the definitions in the time domain, we define $\boldsymbol{s}(\omega_b, n) = [s_1(\omega_b, n), \ldots, s_N(\omega_b, n)]^T$, $\boldsymbol{x}(\omega_b, n) = [x_1(\omega_b, n), \ldots, x_N(\omega_b, n)]^T$, and $\boldsymbol{y}(\omega_b, n) = [y_1(\omega_b, n), \ldots, y_N(\omega_b, n)]^T$, in the frequency domain.

Subsequently, at each moment we use only the current sample block to update the estimation in the prior sample block to obtain a better estimate. In other words, a small number of samples are used at each estimation step of separation parameters, and the sample blocks are different at different estimation steps also. This is different from the conventional batch-type algorithms in which entire samples are used in each learning step; this is the main reason why these types of approaches cannot be used for real-time processing. Therefore, the position of observable sample blocks should be considered as a variable in our estimation. Moreover, we expect to perform CBSS based on a criterion in the frequency domain. Accordingly, we express the criterion to be optimized as the cost function $J(\omega_b, n)$. Here, we treat signals in a block unit because we need the DTFT to act on each block. Conveniently, if we set $n = 1$ for the initial sample block, n is also equal to the number of sample blocks. Note that the separation matrix $\boldsymbol{W}(\omega_b, n)$ is also n-dependent.

B. Criterion for CBSS

In this paper, we consider a modified correlation matrix, defined as

$$\boldsymbol{R}_y(\omega_b, n) \equiv \sum_{k=1}^{n} \beta(n,k) \boldsymbol{\phi}(\boldsymbol{y}(\omega_b, k)) \boldsymbol{\phi}\left(\boldsymbol{y}^H(\omega_b, k)\right) \tag{5}$$

which is a weighted correlation matrix of the function valued CBSS output. Here, $\beta(n,k)$ is the weighting or what we call the forgetting factor. The factor must have the property that $0 < \beta(n,k) \leq 1$ for $k = 1, 2, \ldots, n$. This factor is intended to ensure that samples in the distant past are gradually "forgotten" in order to allow the possibility of following the statistical variations of the observable samples, when the CBSS operates in a nonstationary environment. This paper uses a special form of weighting factor, the exponential forgetting factor, defined by

$$\beta(n,k) = \lambda^{n-k} \tag{6}$$

for $k = 1, 2, \ldots, n$, where λ is a positive constant close to, but less than 1 (typically $\lambda = 0.95$).

The function $\boldsymbol{\phi}(\cdot)$ is defined as

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{y}([\omega_1, \ldots, \omega_L], n)) = [&\boldsymbol{\phi}(y_1([\omega_1, \ldots, \omega_L], n)), \ldots, \\ &\boldsymbol{\phi}(y_M([\omega_1, \ldots, \omega_L], n))]^H\end{aligned} \tag{7}$$

where

$$\begin{aligned}&\boldsymbol{\phi}(y_i([\omega_1, \ldots, \omega_L], n)) \\ &\quad = \mathrm{DTFT}(\boldsymbol{P}_K(\mathrm{IDTFT}(y_i([\omega_1, \ldots, \omega_L], n))))\end{aligned} \tag{8}$$

for $i = 1, \ldots, M$. Here $\boldsymbol{P}_K(\cdot)$ is another function defined as

$$\begin{aligned}&\boldsymbol{P}_K([y(L(n-1)+1), \ldots, y(L(n-1)+L)]) \\ &\quad = [y(L(n-1)-K+1), \ldots, y(L(n-1)), \\ &\qquad y(L(n-1)+1), \ldots, y(L(n-1)+L)].\end{aligned} \tag{9}$$

This means that the operation extends K components in the past from the original vector, keeping the original L components unchanged. If past samples are not long enough, the necessary number of 0-valued elements are added. Consequently, we define $\boldsymbol{P}_K^{-1}(\cdot)$, the inverse of function of $\boldsymbol{P}_K(\cdot)$, as

$$\begin{aligned}&\boldsymbol{P}_K^{-1}([y(L(n-1)-K+1), \ldots, y(Ln)]) \\ &\quad = [0_1, 0_2, \ldots, 0_K, y(L(n-1)+1), y(L(n-1)+2), \\ &\qquad \ldots, y(L(n-1)+L)].\end{aligned} \tag{10}$$

Note that because of the function $\boldsymbol{P}_K$, in (7), the length of DTFT differs from the length of the IDTFT. Indeed, the length of the DTFT is $T = L+K$, while the length of the IDTFT is L. Based on this fact, we have

$$\boldsymbol{\phi}(y_i([\omega_1, \ldots, \omega_L], n)) = y_i([\omega_1, \ldots, \omega_T], n). \tag{11}$$

We term $\boldsymbol{R}_y(\omega_b, n)$ the modified correlation matrix because of the function $\boldsymbol{\phi}$. Notice that the function $\boldsymbol{\phi}(\cdot)$ is relevant to all frequency bins. $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{P}_K(\cdot)$ are introduced for performing the overlap-and-save.

From (4), (7) and (9), we obtain

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{y}([\omega_1, \ldots, \omega_L], n)) = \boldsymbol{\phi}(&\boldsymbol{W}([\omega_1, \ldots, \omega_L], n)) \\ &\cdot \boldsymbol{x}([\omega_1, \ldots, \omega_T], n)\end{aligned} \tag{12}$$

if we input the observation with block length T. Note that the function $\boldsymbol{\phi}(\cdot)$ can also operate on the matrix $\boldsymbol{W}$ defined by $(\boldsymbol{\phi}(\boldsymbol{W}[\omega_1, \ldots, \omega_L], n))_{ij} = \boldsymbol{\phi}(\boldsymbol{W}_{ij}[\omega_1, \ldots, \omega_L], n)$ for $i, j = 1, \ldots, N$, since the length of each matrix component is L. Since $\boldsymbol{W}(\omega_b, n)$ is unknown before the CBSS, $\boldsymbol{P}_K(\boldsymbol{W}(\omega_b, n))$ just means that we set the length of the separating matrix to be $L+K$, although only the last L samples are acted on by the separation matrix to obtain output. Notice when $\boldsymbol{P}_K(\cdot)$ operates on a matrix, its action is on each component vector defined by (9).

By substituting (12) into (5), we obtain

$$\boldsymbol{R}_y(\omega_b, n) = \boldsymbol{P}_K(\boldsymbol{W}(\omega_b, n)) \boldsymbol{R}_x(\omega_b, n) \boldsymbol{P}_K\left(\boldsymbol{W}^H(\omega_b, n)\right) \tag{13}$$

where

$$\boldsymbol{R}_x(\omega_b, n) \equiv \sum_{k=1}^{n} \lambda^{n-k} \boldsymbol{x}(\omega_b, k) \boldsymbol{x}^H(\omega_b, k). \tag{14}$$

Although acoustic signals are not usually generated by ergodic processes, our experiments show that estimating the correlation matrix by (14) still can provide good separation results. The reason is that we provide CBSS outputs via a normalized separating matrix, which is irrelevant to its absolute value [see (24)].

Another purpose for introducing the function $\boldsymbol{P}_K(\cdot)$ is to solve the so-called permutation problem. The effect of the function $\boldsymbol{P}_K(\cdot)$ on the separation filters is to convolve them with a sinc function that is T/Q times wider than the sampling rate. This makes the filter smoother in the frequency domain. Because of this fact, if we choose K such that $T \gg L$, we can solve the permutation problem. In some respects, this is similar to the approach proposed in [20]. With the approach used in [20], after computation of $\boldsymbol{W}$ in each step, a constraint is imposed such that $\boldsymbol{W}(\tau) = 0$ for $\tau > L$, where $L \ll T$. Although similar, in our case, we expressed this constraint as an indirect effect, i.e., as a byproduct of the overlap-and-save strategy. By

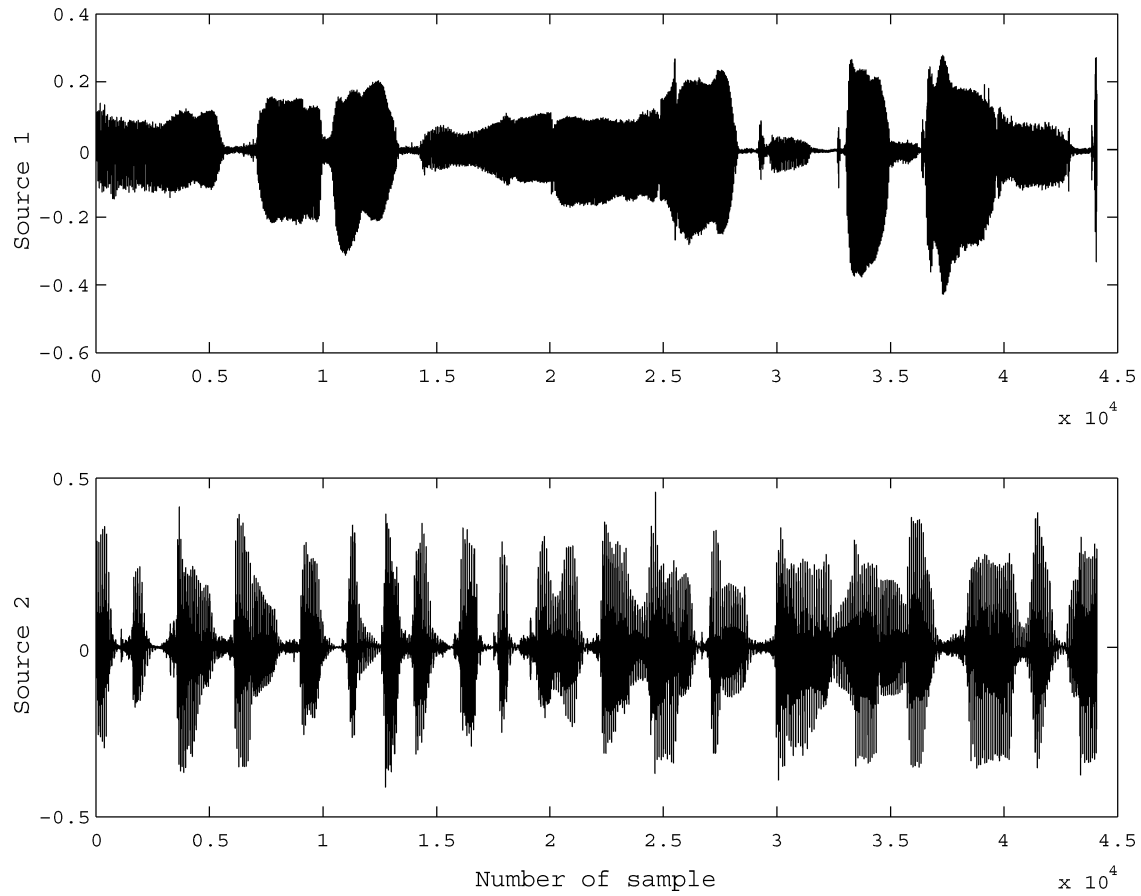

Fig. 1. Two independently generated examples of audio source signals for testing the validity of the separation criteria (15) and (18).

contrast, Parra and Spence expressed the constraint as an artificially imposed condition. In summary, the overlap-and-save strategy has four effects, which are necessary for a real-time CBSS algorithm: (a) it converts circular convolution into linear convolution [21]; (b) it solves the permutation problem, as explained above; (c) it solves the permutation-switching problem of separating matrices from one block to the next block if there is sufficient overlap; and (d) it introduces time-lagged correlation in addition to zero-lag correlation to produce a better separation in each frequency bin, as explained in Section II-D.

The problem of BSS can now be formulated to find $\boldsymbol{W}(\omega_b, n)$ such that the modified correlation matrix will be diagonalized after convergence, i.e.,

$$\boldsymbol{R}_y(\omega_b, n) = \boldsymbol{\Lambda}(\omega_b, n), \quad \forall \omega_b, n \tag{15}$$

where $\boldsymbol{\Lambda}(\omega_b, n) \equiv \mathrm{diag}(\Lambda_i(\omega_b, n))$ is a diagonal matrix. This is equivalent to a problem that needs to find $\boldsymbol{W}(\omega_b, n)$, such that it minimizes the cost function

$$J(\omega_b, n) = \sum_{i \neq j} \left| [\boldsymbol{R}_y(\omega_b, n)]_{ij} \right|^2 \tag{16}$$

for $b = 1, \ldots, T$ and $n < \infty$.

To test the validity of this idea, we checked the weighted cross correlation between two arbitrarily chosen audio sources that were generated independently. A pair of such source examples is shown in Fig. 1. Suppose CBSS processing is performed perfectly, then we would express the outputs to be identical to the two original sources, i.e., $\boldsymbol{y}(\omega_b, n) = \boldsymbol{s}(\omega_b, n)$. In this case, we must verify that the off-diagonal components of $\boldsymbol{R}_y(\omega_b, n)$ become sufficiently small in each frequency bin, as indicated by the criterion (15). Fig. 2 shows the weighted cross correlation in every frequency bin for the examples. Here, the block length of the DTFT is 4096. We can see that in most bins the off-diagonal components are indeed negligible, such that the condition is approximately satisfied. However, it does not hold anymore for very low-frequency bins. That is, to some extent, overlearning[2] occurs if we use (15) as the separation criterion. To temper this overlearning effect, we must implement the following.

We define a normalized correlation matrix $\overline{\boldsymbol{R}}_y(\omega_b, n)$ by

$$\left[\overline{\boldsymbol{R}}_y(\omega_b, n) \right]_{ij} = \frac{[\boldsymbol{R}_y(\omega_b, n)]_{ij}}{\Lambda_i(\omega_b, n)} \tag{17}$$

then after convergence to a global minimum we have

$$\overline{\boldsymbol{R}}_y(\omega_b, n) = \boldsymbol{I}. \tag{18}$$

Now, let us determine the weighted cross correlation $[\overline{\boldsymbol{R}}_y(\omega_b, n)]_{i \neq j}$ between the independently generated audio sources shown in Fig. 1. Fig. 3 shows these results. We see that the weighted cross correlations are indeed near zero for

[2]Any two perfectly estimated acoustic sources may correspond to a minimum point of cross correlation between them via various mixing parameters, but not necessarily to a 0 valued point, although it is approximately correct in most situations. Then, if we enforce the cross-correlation vanishing, the estimated source might be close to but different from the sources. The term "overlearning" denotes such an effect.

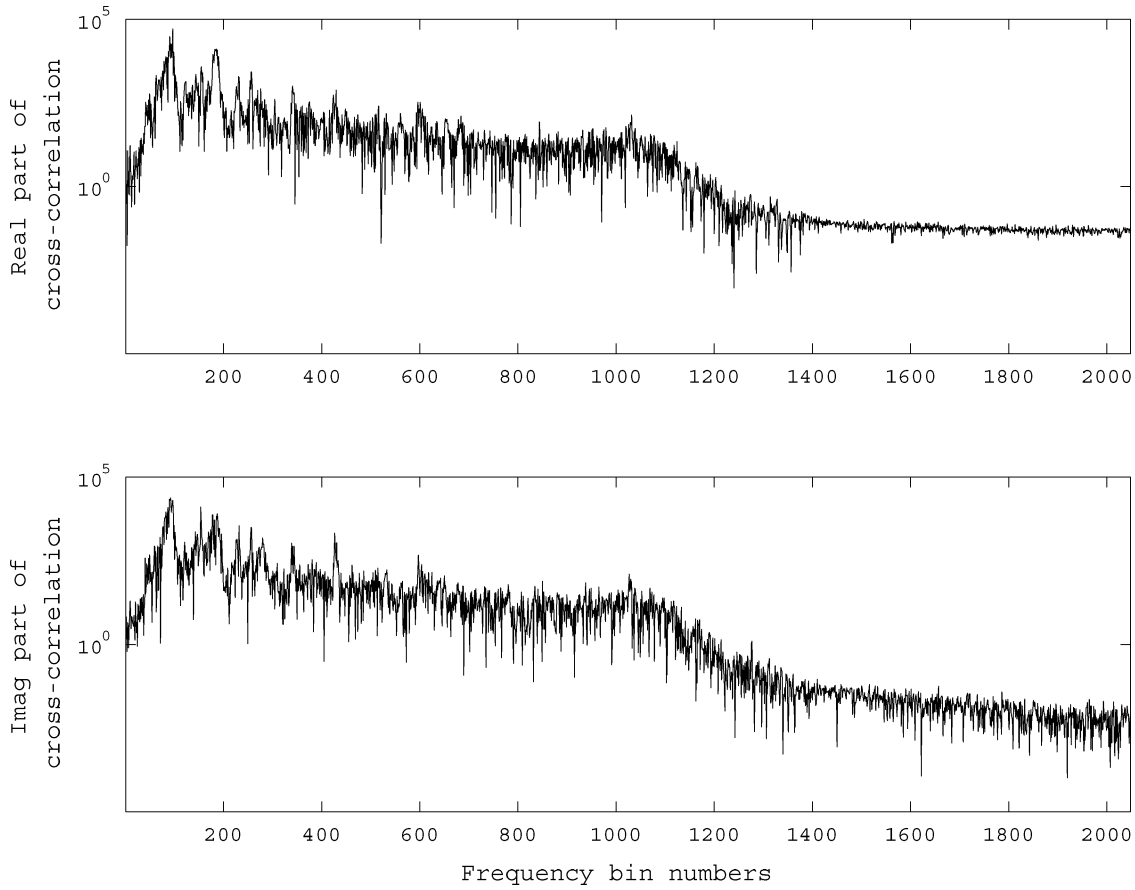

Fig. 2. Weighted cross correlations in frequency bins (signals: the audio sources shown in Fig. 1).

independently generated sources in all frequency bins. In this case, the averaged value of the cost function (16) over b was 9.80×10^{-4} or -30.10 dB, which is very small comparing with that of $|\Lambda_i(\omega_b, n)|^2$, $\forall i$. We have verified that this is common for most speech signals, if they are generated independently. For another pair of audio examples, the value of the cost function was 1.9167×10^{-4} or -37.17 dB. Because this value is not exactly zero, overlearning will still occur when (18) is used as a separation criterion. However, this value is so small that we can regard it as zero approximately. Since all CBSS algorithms only separate audio sources approximately, the approach is still useful if the approximation does not harm the separation too much. If the crosstalk remaining in the outputs due to overlearning is less than the crosstalk remaining in the outputs due to underlearning associated with other problems (e.g., local optimal points), the approach associated with overlearning is still applicable.

C. Method for Near Real-Time CBSS Processing

If we define a normalized separation matrix by

$$\overline{\boldsymbol{W}}(\omega_b, n) = \boldsymbol{\Lambda}(\omega_b, n)^{-\frac{1}{2}} \boldsymbol{W}(\omega_b, n) \tag{19}$$

then (18) can also be rewritten as

$$\boldsymbol{P}_K \left(\overline{\boldsymbol{W}}(\omega_b, n) \right) \boldsymbol{R}_x(\omega_b, n) \boldsymbol{P}_K \left(\overline{\boldsymbol{W}}^H(\omega_b, n) \right) = \boldsymbol{I}, \forall \omega_b, n. \tag{20}$$

In deriving this equation, we used the property that matrix $\boldsymbol{\Lambda}(\omega_b, n)$ is Hermitian, i.e., $\boldsymbol{\Lambda}(\omega_b, n) = \boldsymbol{\Lambda}^H(\omega_b, n)$, because it is a diagonal, real matrix.

We now will discuss how to solve (20) to give $\overline{\boldsymbol{W}}(\omega_b, n)$. However, there is a problem when one tries to solve $\boldsymbol{W}(\omega_b, n)$ from (19), even if $\overline{\boldsymbol{W}}(\omega_b, n)$ has been given. One cannot uniquely determine matrix $\boldsymbol{\Lambda}(\omega_b, n)$, since the sources are unknown and so $\boldsymbol{\Lambda}(\omega_b, n)$ is also unknown. There exists an ambiguity in CBSS, which states that one cannot determine the absolute powers of the outputs. Our problem is profoundly related to this ambiguity.

Because the components of matrix $\boldsymbol{\Lambda}(\omega_b, n)$ represent only power distributions of sources, one can predesignate distributions for them. This predesignation does not affect the quality of the BSS. However, this approach will change the spectral envelopes of the signals, making them either greater or lesser than those of the original sources. In this paper, we propose a new approach for solving this problem.

Because, from (4), the ith output is given by

$$y_i(\omega_b, n) = \sum_{k=1}^{N} W_{ik}(\omega_b, n) x_k(\omega_b, n) \tag{21}$$

we may consider the row vector $\boldsymbol{W}_i \equiv [W_{i1}, \ldots, W_{iN}](\omega_b, n)$ as the ω_b-component of a generalized filter.

If we impose a normalization condition on vector W_i to make its norm constant in each frequency bin, the filter will become an all-pass filter. Such a filter will not affect the power distributions,

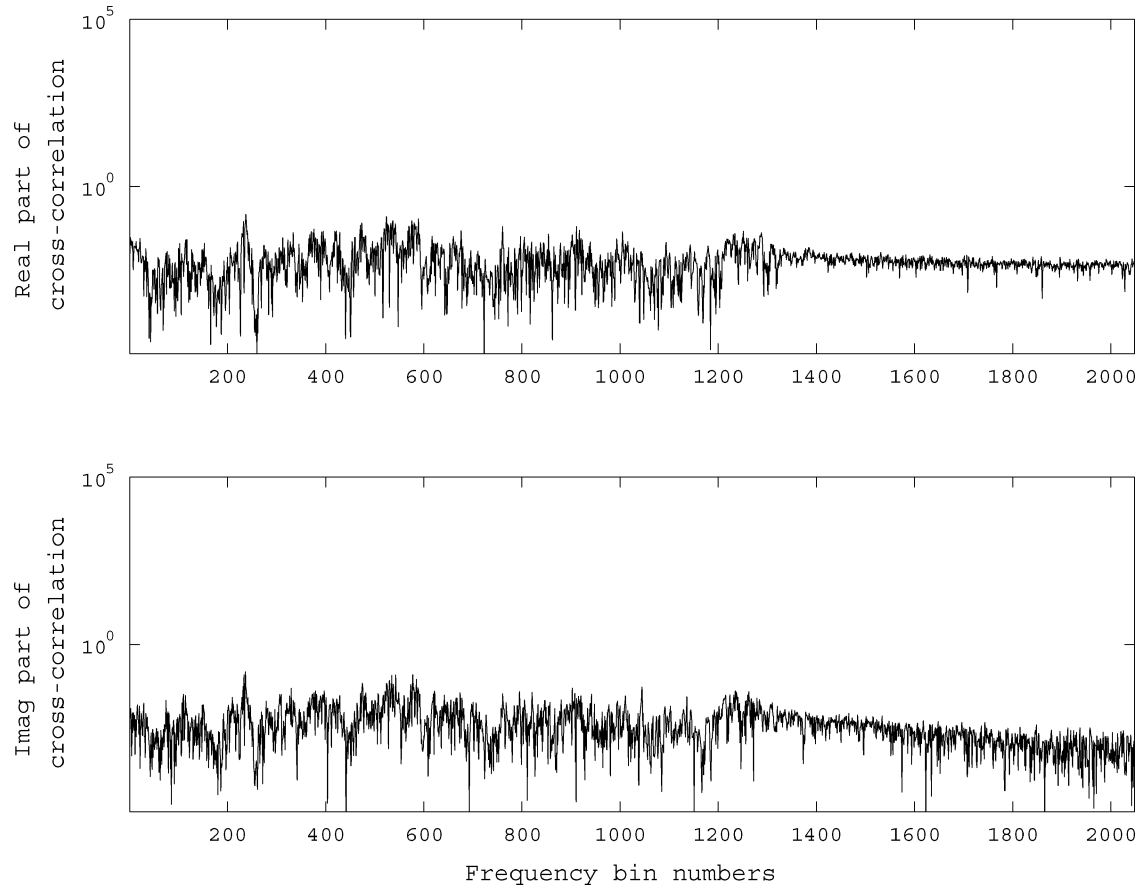

Fig. 3. Normalized weighted cross correlations in frequency bins (signals: the audio sources shown in Fig. 1).

and therefore neither will it affect the spectral envelopes of the signals. We can also use such a constraint to make the outputs unrelated to $\mathbf{\Lambda}(\omega_b, n)$. The powers of outputs will also become definite due to the constraint; i.e., after the normalization, we no longer have the power ambiguity. With CBSS, note that we had the power ambiguity, that is, the scale indeterminacy. That is why we can choose the normalization as a constraint to remedy the ambiguity. In this way, instead of $y_i(\omega_b, n)$ given by (21), we want to obtain the output $y_i(\omega_b, n)$ given by

$$y_i(\omega_b, n) = \sum_{k=1}^{N} \frac{W_{ik}(\omega_b, n)}{\|\boldsymbol{W}_i(\omega_b, n)\|} x_k(\omega_b, n) \tag{22}$$

where $\|\cdot\|$ denotes the L_2 norm operation on a row vector.

Using (19), it is easy to show that

$$\frac{(W_{i1}, \ldots, W_{iN})(\omega_b, n)}{\|\boldsymbol{W}_i(\omega_b, n)\|} = \frac{(\overline{W}_{i1}, \ldots, \overline{W}_{iN})(\omega_b, n)}{\left\|\overline{\boldsymbol{W}}_i(\omega_b, n)\right\|} \tag{23}$$

Therefore, (22) becomes

$$y_i(\omega_b, n) = \sum_{k=1}^{N} \frac{\overline{W}_{ik}(\omega_b, n)}{\left\|\overline{\boldsymbol{W}}_i(\omega_b, n)\right\|} x_k(\omega_b, n). \tag{24}$$

The merit of using (24) instead of (21) is that it is independent of $\mathbf{\Lambda}(\omega_b, n)$. That is, once we have obtained $\overline{W}_{ik}(\omega_b, n)$ by solving (20), we can substitute it into (24) to calculate the output without considering $\mathbf{\Lambda}(\omega_b, n)$.

D. Related Discussion for the Proposed Criterion and Method

We have proposed a CBSS algorithm that stems from the following principle: For the function given by (9), the separating network ought to diagonalize the matrix $\boldsymbol{R}_y(\omega_b, n)$ in each frequency bin. As is well known, signals that are not cross correlated need not be independent. Here, we discuss theoretical considerations providing evidence to support the validity of the proposed criterion for separation. In particular, we address the need for the so-called simultaneous diagonalizations of several correlation matrices for cases in which the sources are nonstationary.

Parra and Spence [20] proposed an approach based on simultaneous diagonalization of the correlation matrices for several different time windows.[3] As is well known, however, a more efficient criterion is one that simultaneously diagonalizes the time-lagged cross correlation, in addition to the zero-lag cross correlations [22], [23]. In fact, if one makes only the zero-lag spatial decorrelation, it is equivalent to a principal component analysis (PCA) that cannot sufficiently separate sources from their instantaneous mixtures. Therefore, one cannot expect a good overall separation by applying PCA or standard spatial decorrelation in each frequency bin. Indeed, Ikram and Morgan evaluated the algorithm and found that the separation performances were insufficient in some reverberant environments [24]. Although they attributed their finding to a permutation

[3]This is the zero-lag, that is, the equal-time correlation estimated at different times, which might make the estimation more accurate.

problem, another reason could be insufficient separation in each frequency bin. The permutation problem should be considered only when the separation in every frequency bin is sufficient good. In this paper we want to consider cross correlations with and without time lags simultaneously. However, the method developed in [22] and [23] is rather difficult to implement as part of a RLS-type of algorithm, which is a main objective in this paper. Instead, we have introduced the overlapping sample blocks for considering both the zero-lag and time-lagged cross correlations. In the Appendix, we show that our cost function really introduces time-lagged cross correlations.

However, even if a cross correlation defined as (A.38) vanishes, it is not necessary that both the zero-lag correlation and the time-lagged correlations vanishes simultaneously, since not all matrices are positive definite. For solving this problem, we have proposed a novel approach for the simultaneous diagonalization, which is based on an iterative processing method, as explained in the Appendix.

We may also reconsider this proposed approach from another viewpoint. As concluded by Cichocki *et al.* [25]–[27], one can successfully perform BSS even in cases in which only the zero-lag correlation is considered and in which the generalized correlation matrix is constructed by nonlinear transformations of signals. The nonstationarity of sources also becomes unnecessary. In some sense, our approach is similar to a diagonalization of such a nonlinear functioned correlation matrix. The function $\phi(\cdot)$, although being linear, has a similar effect as the nonlinear functions applied in [25]–[27].

III. Exponentially Weighted Explicit CBSS Algorithm

We might call (20) a "normal equation" of CBSS, analogous to the normal equation of the RLS algorithm in the theory of adaptive filters [21]. The CBSS problem can now be thought of as one in which (20) needs to be solved. In [18], we proposed a recursive approach for solving the normal equation. In this section, we present a different method to solve (20), which makes online processing quite easy. First, (20) can be written as

$$\boldsymbol{P}_K\left(\overline{\boldsymbol{W}}^H(\omega_b,n)\right)\boldsymbol{P}_K\left(\overline{\boldsymbol{W}}(\omega_b,n)\right)=\boldsymbol{R}_x^{-1}(\omega_b,n). \quad (25)$$

From (25) we can solve $\overline{\boldsymbol{W}}(\omega_b,n)$ by

$$\overline{\boldsymbol{W}}(\omega_b,n)=\boldsymbol{P}_K^{-1}\left(\boldsymbol{V}(\omega_b,n)\boldsymbol{D}(\omega_b,n)^{-\frac{1}{2}}\boldsymbol{V}(\omega_b,n)^H\right) \quad (26)$$

where $\boldsymbol{D}(\omega_b,n)=\mathrm{diag}(\lambda_1(\omega_b,n),\ldots,\lambda_N(\omega_b,n))$ denotes a diagonal matrix of eigenvalues $\lambda_i(\omega_b,n)$ for $i=1,2,\ldots,N$, and $\boldsymbol{V}(\omega_b,n)$ denotes the eigenvectors of the matrix $\boldsymbol{R}_x(\omega_b,n)$. Although the square root is multivalued, here we have chosen the principal value as our solution. Because $\boldsymbol{R}_x(\omega_b,n)$ is a Hermitian matrix, the eigenvalues are real. Therefore, the principal value of the square root in (26) is uniquely defined.

From (14), it is easy to show that the nth correlation matrix $\boldsymbol{R}_x(\omega_b,n)$ is related to the $n-1$th correlation matrix $\boldsymbol{R}_x(\omega_b,n-1)$ by

$$\boldsymbol{R}_x(\omega_b,n)=\lambda\boldsymbol{R}_x(\omega_b,n-1)+\boldsymbol{x}(\omega_b,n)\boldsymbol{x}^H(\omega_b,n) \quad (27)$$

where the matrix product $\boldsymbol{x}(\omega_b,n)\boldsymbol{x}^H(\omega_b,n)$ plays the role of a "correction" term in the updating operation.

Although one can take any value of T such that $T\gg L$, for convenience of realization the algorithm, we assume $T=DL$ (or say, $K=(D-1)L$), where D is a positive integer. A typical value for D is from 4 to 8.

As described in Section II, to simultaneously diagonalize the correlation matrix, we use the sequence processing method. A single value-lagged decorrelation, including also the 0-lagged (i.e., the equal-time one), obtains only the solution (26) for a set of matrices $\boldsymbol{R}_x(\omega_b,n)$, for all n and a sequence of single valued lags. That is, for a input block, we require both of updating

$$\overline{\boldsymbol{W}}(\omega_b,n)=\overline{\boldsymbol{W}}(\omega_b,n-1)\boldsymbol{P}_K^{-1}\left(\boldsymbol{R}_x^{-\frac{1}{2}}(\omega_b,n-1)\boldsymbol{R}_x^{-\frac{1}{2}}(\omega_b,n)\right) \quad (28)$$

with respect to n, and updating

$$\overline{\boldsymbol{W}}^{(l)}(\omega_b,n)=\overline{\boldsymbol{W}}^{(l-1)}(\omega_b,n)\cdot\boldsymbol{P}_K^{-1}\left(\left(\boldsymbol{R}_x^{(l-1)}(\omega_b,n)\right)^{-\frac{1}{2}}\left(\boldsymbol{R}_x^{(l)}(\omega_b,n)\right)^{-\frac{1}{2}}\right) \quad (29)$$

with reqspect to $l=1,\ldots,D-1$, for a fixed n. Here $\boldsymbol{R}_x^{(l)}(\omega_b,n)$ is defined in the same way as (A.42), but replacing $\bar{\boldsymbol{y}}(\omega_b,k)$ by $\bar{\boldsymbol{x}}(\omega_b,k)$. $\overline{\boldsymbol{W}}(\omega_b,n)$ relates to $\overline{\boldsymbol{W}}^{(l)}(\omega_b,n)$ by

$$\overline{\boldsymbol{W}}(\omega_b,n)=\overline{\boldsymbol{W}}^{(D-1)}(\omega_b,n). \quad (30)$$

It is easy to prove that (28) and (29) solve (26), if $\overline{\boldsymbol{W}}(\omega_b,n-1)$ and $\overline{\boldsymbol{W}}^{(l-1)}(\omega_b,n)$ are solutions. Note that the updating in (28) and (29) is not stochastic learning but an iteratively performing algorithm of joint diagonalization of correlation matrices with different time lags.

Fig. 4 shows a block diagram for solving (26) and (27) recursively. The block "Update to $\boldsymbol{R}(\omega_b,n-1)$" denotes a calculation of the second term on the right side of (27). In the figure, we have ignored the steps for the DTFT on the input signals and the step for the IDTFT on the output signals. Here, we adopt the overlap-and-save method for the real-time DTFT-IDTFT processing step.

Although it is unnecessary, from here on, we chose $K=3L$ (or $K=7L$), which means there is an overlap of $3L$ (or $7L$) samples between neighboring blocks.

The initial condition for the recursive processing is $\boldsymbol{R}_x(\omega_b,n)=\boldsymbol{I}$, for $n\leq 0$.

From Fig. 4, we see that there is a very large difference between the construction of our CBSS algorithm and that of the conventional gradient-type algorithm. For the gradient-type of algorithm, some parameters related to the output signals usually need to be estimated, and then the results of these estimates are fed back to update the separation matrix of filters. With our recursive CBSS algorithm, however, this kind of feedback is unnecessary. All of the estimates that are exploited to update the separation matrix are on the input signals only.

IV. Simulations and Experiments

To evaluate our proposed recursive BSS system, we conducted several simulations and experiments using various environments. We obtained very different results depending on reverberation characteristics of the room, stationarity of the source signals, the numbers of sources and microphones,

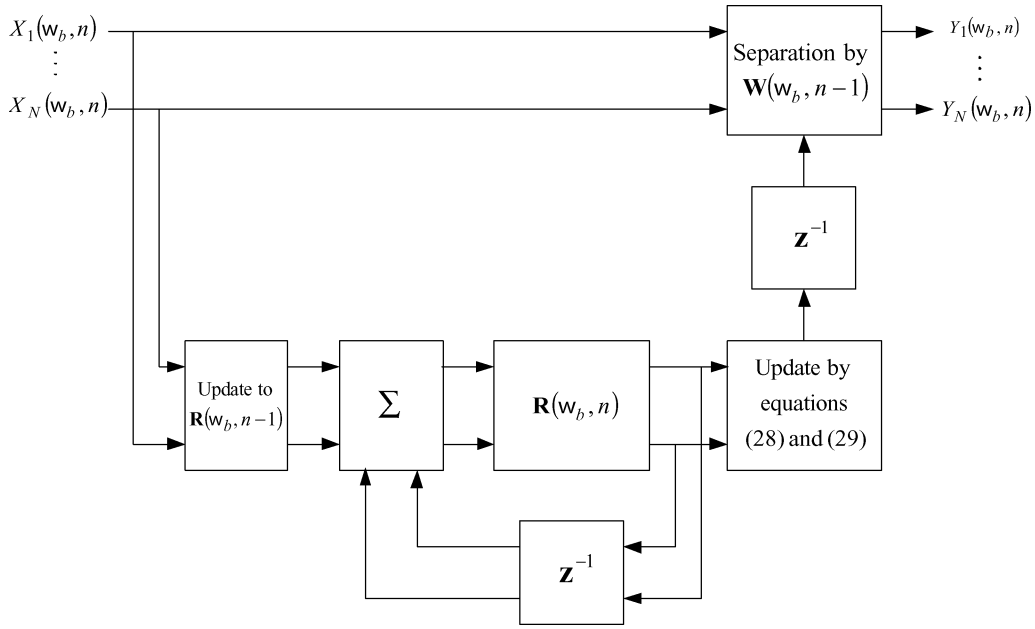

Fig. 4. Block diagram for recursive CBSS.

background noise, and other factors. Here, we report some results of these simulations and experiments with recordings or real-time inputting of voices in realistic environments, such as offices and vehicles.

For adapting the same cost function (16), we can choose the gradient-based algorithm or the recursive algorithm proposed in this paper. As explained in the introduction, we propose using the recursive algorithm for attaining faster convergence and for attaining convergence to a lower value of the cost function than what would be attained using a conventional gradient-based algorithm. In this section, we present several results from simulations and experiments to demonstrate that this is indeed the case. The gradient-based algorithm in [15] has been chosen for comparison.

In the conventional gradient-based algorithms, there is an algorithm parameter called "learning step-size" that must be determined beforehand. Instead of learning step-size, the proposed algorithm in this paper contains a different parameter, called the forgetting constant λ, in the modified correlation matrix that also must be determined beforehand. The choice of these parameters will affect the performance of the algorithm. To find which algorithm is more efficient, as a first step, we established optimal parameter values for each algorithm with the aim of achieving the best convergence and performance. Of course, these optimum values also depend on the signal sources and mixing environments. During the convergence process, we evaluated the value of the cost function and plotted convergence curves. Since there exist some fluctuations in the convergence curves, we used average values in several (e.g., 100) iterations to determine which parameter produces faster convergence. It should be noted that, although we use different algorithms, we employ the same cost function. Using their optimal parameters, we compared the performance of the two types of algorithms. In this section, we present the obtained simulation and experimental results for such comparisons.

Note that in all simulations and experimental results, the iteration number is also equal to n, the block number, since we updated the estimation of the correlation matrix once for each input signal block. Note also that there is no learning in our proposed algorithm; an iteration only appears in updating the estimation of the correlation matrix.

A. Simulation Results for Separation of I.I.D. Sources

For simplicity, as a first step, we checked the separation performance for sources that are I.I.D.. The statistics of the sources were chosen to be super-Gaussian, since a speech source is usually distributed as such. In this simulation, we used a known matrix to mix two sources and to produce two observations. To evaluate the performance of the proposed algorithm, we employed multichannel intersymbol interference (ISI) [28]–[30], denoted by $\mathrm{M_{ISI}}$, as a criterion index

$$\mathrm{M_{ISI}} = \sum_{i=1}^{N} \frac{\sum_{j=1}^{N}\sum_{t=0}^{L-1}|g_{ij}(t)| - \max_{t,j}|g_{ij}(t)|}{\max_{t,j}|g_{ij}(t)|} + \sum_{j=1}^{N} \frac{\sum_{i=1}^{N}\sum_{t=0}^{L-1}|g_{ij}(t)| - \max_{t,i}|g_{ij}(t)|}{\max_{t,i}|g_{ij}(t)|} \quad (31)$$

where $g_{ij}(t) = \sum_{k=1}^{N} W_{ik}(t) * A_{kj}(t)$ is the global transfer function combining the mixing and separating processes.

In Fig. 5, $\lambda = 0.95$ for our proposed recursive algorithm and $\mu = 0.01$ for the gradient CBSS algorithm. The filter lengths in the FIR filter matrix was set to 32, and the length L of the DTFT was 128. The sample block size was 128, and the number of shifting samples between neighboring blocks was 32. The curve shows an average result of 100 simulation runs. In each run, the mixing matrix and the initial separating matrix were randomly chosen.

The results presented in Fig. 5 clearly show the superior convergence rate of the recursive CBSS over its counterpart, the gradient CBSS algorithm in [15].

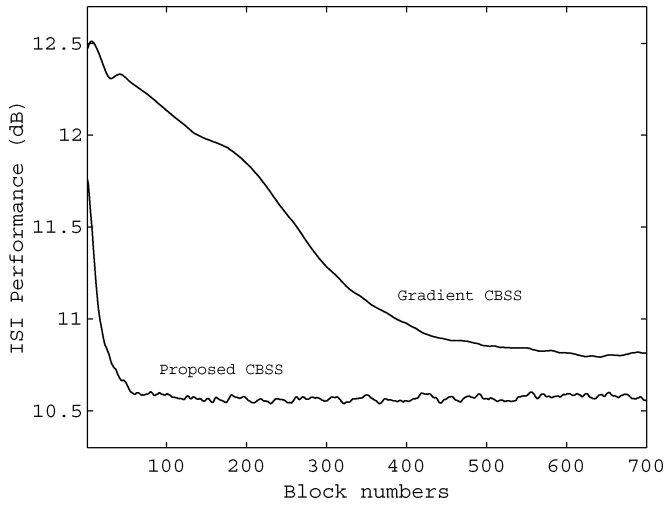

Fig. 5. Learning curves for the proposed recursive CBSS algorithm and gradient-based CBSS algorithm.

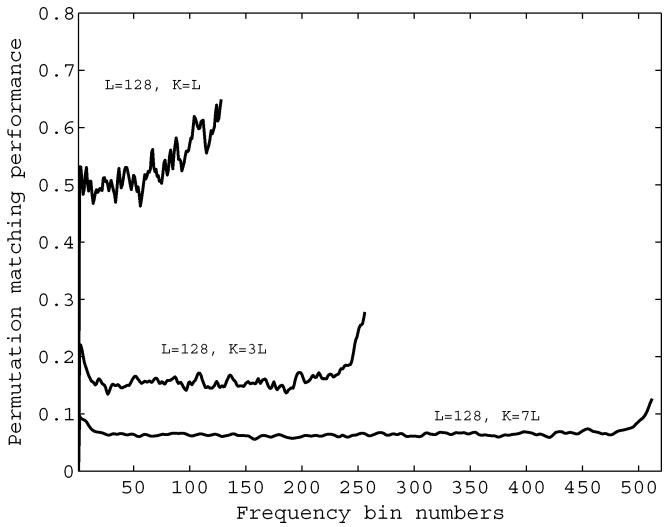

Fig. 6. Permutation matching performance between bins at the final output block.

B. Simulation Results for the Permutation Mismatching Problem

This simulation was performed to evaluate the efficiency of our approach for solving the permutation mismatching problem. In the literature, this is also known as the permutation problem or permutation indeterminacy. In each simulation, we randomly designated a mixing matrix of FIR filters and determined $\boldsymbol{A}(\omega_b, n)$. Then we ran CBSS to produce $\boldsymbol{W}(\omega_b)$ and calculate

$$\boldsymbol{H}(\omega_b, n) = \boldsymbol{W}(\omega_b, n)\boldsymbol{A}(\omega_b, n). \tag{32}$$

After CBSS, $\boldsymbol{H}(\omega_b, n)$ should be a diagonal matrix, or a permuted diagonal matrix, for each ω_b. Since different permutations should yield nonzero components at different positions of the matrix

$$\mathrm{PM}(\omega_b, n) = \frac{\|\boldsymbol{H}(\omega_{b+1}, n) - \boldsymbol{H}(\omega_b, n)\|_F}{\|\boldsymbol{H}(\omega_{b+1}, n)\|_F} \tag{33}$$

is large. Here $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. If it is close to 0, the two neighboring bins have the same permutation. Therefore, we can consider $\mathrm{PM}(\omega_b, n)$ to be a permutation mismatching performance index between neighboring bins.

We ran the simulation 100 times and took the average of the permutation mismatching performance index. For each simulation, the mixing matrix was chosen randomly. The results are shown in Fig. 6.

Since K was chosen to be different, for the simulations presented in Fig. 6, the length T of the DTFT is correspondingly different. This explains why the total number of bins is different. From Fig. 6, larger K values produced less mismatching. This is just what we expected. Notice here that, for the case of $L = 128$, $K = L$, the permutation mismatching performance index is always at least 0.5. This does not mean that there is always mismatching. Because we conducted 100 runs and took the average, the large value means that mismatching occurred more often.

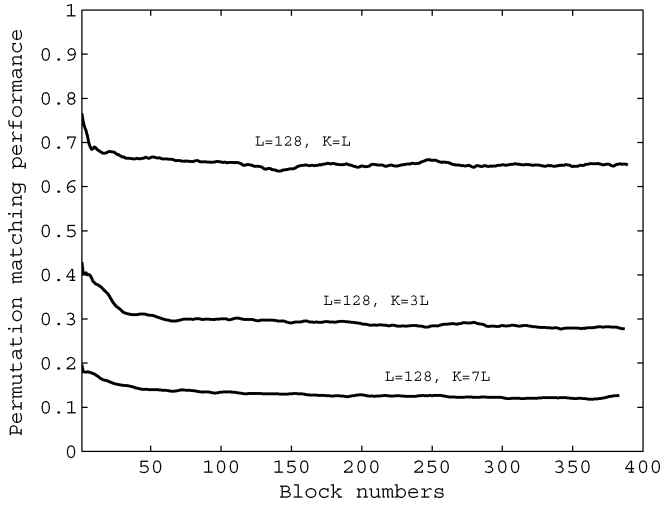

Fig. 7. Permutation matching performance between bin 127 and bin 128.

Fig. 7 shows the performance of permutation matching between bin 127 and bin 128, and Fig. 8 shows the average performance of permutation matching over all bins, which is defined as

$$\mathrm{PM}(n) = \frac{1}{\frac{T}{2}+1} \sum_{b=1}^{\frac{T}{2}+1} \mathrm{PM}(\omega_b, n). \tag{34}$$

From Figs. 7 and 8, we see that our algorithm "converges" to solving the permutation mismatching.

C. Simulation Results for the Permutation Switching Problem

This simulation was conducted to evaluate the efficiency of our algorithm for solving the permutation switching problem. Because our algorithm calculates the separating matrix one time at each block, we must guarantee that outputs are produced with the same order from one block to the next. We consider permutation switching to have occurred when output orders are changed at two neighboring blocks.

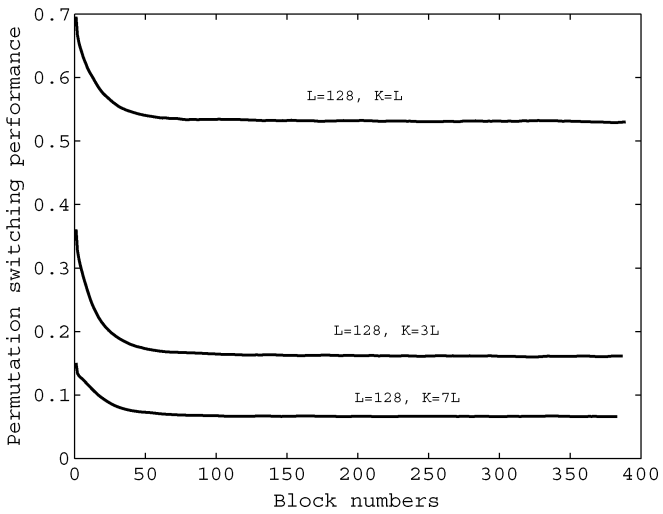

Fig. 8. Average permutation matching performance over all bins.

Similar to the permutation mismatching case discussed above, we can also define a permutation switching performance index by

$$\mathrm{PS}(n,\omega_b)=\frac{\|\boldsymbol{H}(\omega_b,n+1)-\boldsymbol{H}(\omega_b,n)\|_F}{\|\boldsymbol{H}(\omega_b,n+1)\|_F} \tag{35}$$

where the matrix norm is defined the same as in (33). If the value is close to 0, no permutation switching has occurred; if the value is close to 1, permutation switching has occurred.

We can also define the averaged permutation switching performance index over all bins by

$$\mathrm{PS}(n)=\frac{1}{\frac{T}{2}+1}\sum_{b=1}^{\frac{T}{2}+1}\mathrm{PS}(n,\omega_b). \tag{36}$$

Fig. 9 shows the performance of our algorithm for simulations involving permutation switching via frequency bin numbers. Each curve shows the permutation switching performance between the final block and the final block minus one block.

Fig. 10 shows the simulation permutation switching performance via block numbers at bin 128.

Fig. 11 shows the average performance for the permutation switching index over frequency bins via block numbers. As is expected, Fig. 11 is similar to Fig. 10, except for the smoothing effect resulting from averaging over frequency bins.

All of the simulation results are averages of results calculated over 100 runs. From Figs. 10 and 11, we see that our algorithm "converges" to solving the permutation switching.

D. Simulation Results for Separation of Real-Word Benchmark Recordings

Here, two real-world benchmark recordings, [31] and [32], were used to evaluate our proposed recursive CBSS algorithm. The benchmark [31] was recorded in a typical office room. The distance between the speakers and the microphones was about 60 cm and were positioned in square arrangement. The sampling rate was 16.0 kHz. The benchmark [32] was recorded in a room with the following dimensions: $3.4\,\mathrm{m}\times 3.8\,\mathrm{m}\times 5.2\,\mathrm{m}$ (height $\times$ width $\times$ depth). Two speakers (0.80 m apart) were arranged parallel to two microphones (0.58 m apart); the speakers and microphones were 1.2 m apart. The sampling rate was 24.0 kHz.

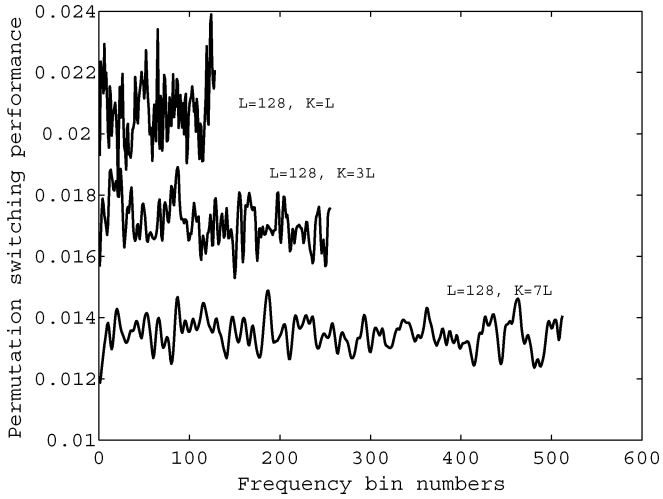

Fig. 9. Permutation switching index of the final and the final minus one block.

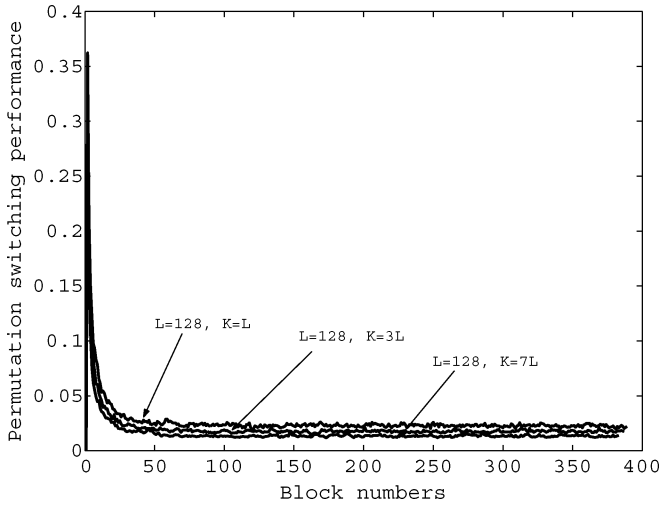

Fig. 10. Permutation switching performance at bin 128.

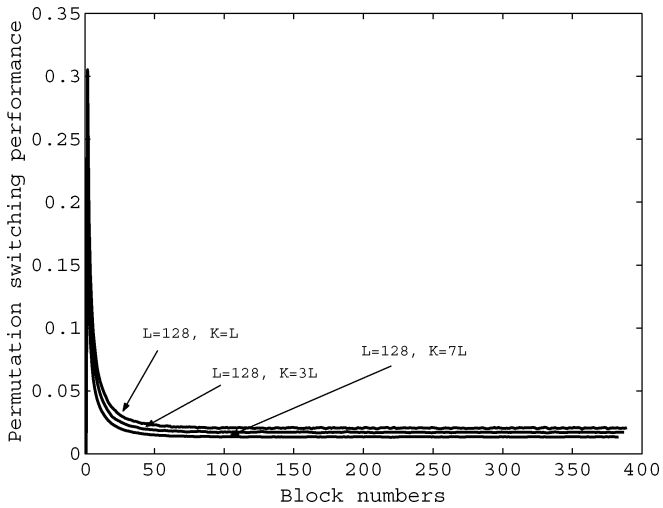

Fig. 11. Average permutation switching performance over all bins.

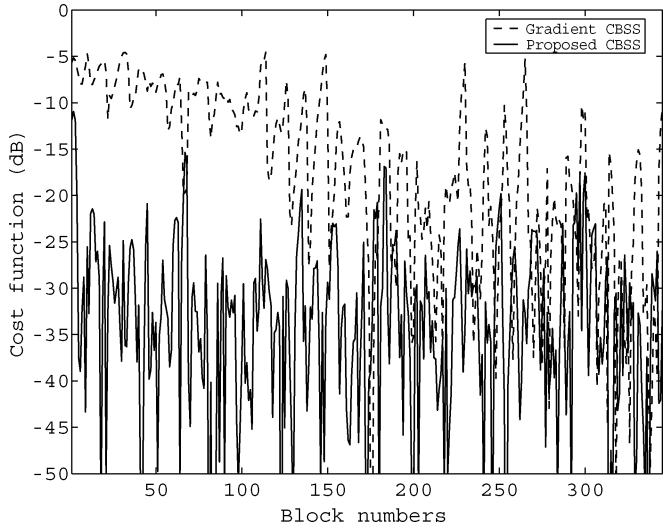

Fig. 12. Learning curves for the proposed recursive CBSS algorithm and the gradient-based CBSS algorithm: benchmark [31].

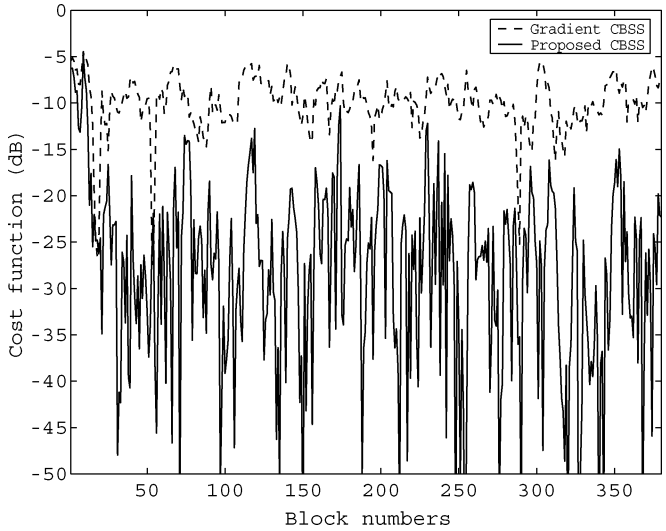

Fig. 13. Learning curves for the proposed recursive CBSS algorithm and the gradient-based CBSS algorithm: benchmark [32].

In these simulations and in the experiment described in the next subsection, because nothing is known about the mixing matrix, we cannot calculate the criterion (31). Instead, we first evaluated the convergence rates of the cost function (16). Figs. 12 and 13 show the learning curves for the algorithm [15], one of the conventional gradient-based algorithms, and for our proposed recursive CBSS algorithm. For the gradient algorithm, we chose the same cost function and the optimum step-size. The results presented in Fig. 12 clearly show the superior convergence rate of the recursive CBSS algorithm over its counterpart, the gradient CBSS algorithm [15]. The results presented in Fig. 13 demonstrate that recursive CBSS converges in almost the same way as the simulation presented in Fig. 12. In contrast, the counterpart hardly converges at all.

For the analysis presented in Fig. 12, $\lambda = 0.95$ for the recursive algorithm and $\mu = 0.01$ for the gradient CBSS algorithms. As explained in the beginning of this section, these are optimum values for the corresponding algorithms. Observation signals rss_mA and rss_mB [31] were used. The sampling rate was 16.0 kHz. The filter lengths in the FIR filter matrix were set to 1024, and the length T of the DTFT was 4096. The sample block size was 4096, and the number of shifting samples L between neighboring blocks was 1024. That is, 75% $(K = 3072 \text{ samples})$ samples overlapped in two consecutive blocks.

The convergence occurred in 10 iterations. Because the algorithm needs one processing iteration for each shift, the convergence time was $10 \times 1024/16\,000 = 0.64$ s. Here we suppose that the hardware calculation was so fast that there was no waiting time between iterations.

For the analysis presented in Fig. 13, $\lambda = 0.95$ for the recursive algorithm and $\mu = 0.01$ for the gradient CBSS algorithm. Fig. 13 shows that the gradient CBSS algorithm does not converge in this case. To determine the optimal value of μ in this case, we chose a value for μ that enables the algorithm obtain a set of outputs with the lowest *averaged* cost function values over 100 iterations. The observation signals "spc2x1" and "spc2x2" [32] were used. The sampling rate was 24.0 kHz. The filter lengths in the FIR filter matrix were set to 1024 and the length T of the DTFT was 4096. The sample block size was 4096, and the number of samples shifting L between neighboring blocks was 1024. Similar to the last simulation, 75% $(K = 3072 \text{ samples})$ of the samples overlapped in two consecutive blocks.

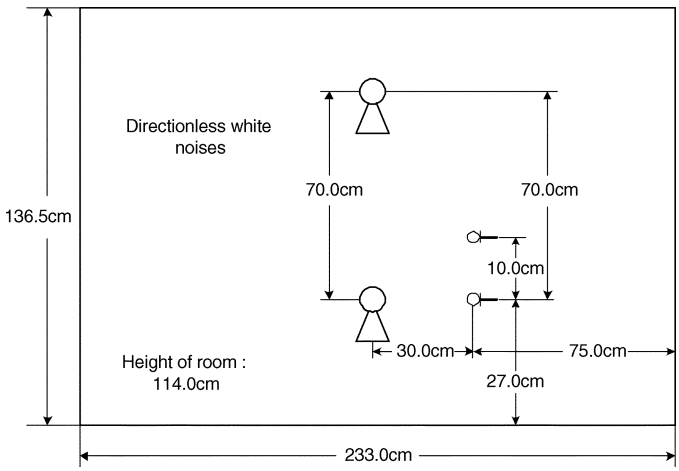

Fig. 14. Schematic of realistic environment used in experimental studies.

Convergence occurred in 30 iterations. Because the algorithm needs one processing iteration for each block, the convergence time was $30 \times 1024/24\,000 = 1.28$ s.

The wild fluctuations in the learning curves presented in Figs. 12 and 13 and in the following figures are due to the subtle changes in the position of the speakers in the recordings and the nonstationarity of the sources.

E. Experimental Results for Separation of Real-World Recordings

Real-time experiments have been implemented both on a Matlab/Simulink model and on a TMS320C6701 Evaluation Module board from Texas Instruments.

Experiments were conducted in a realistic car environment. The car passenger compartment measured 114.0 cm × 136.5 cm × 233.0 cm (height × width × depth), and is schematically depicted in Fig. 14. Two persons read

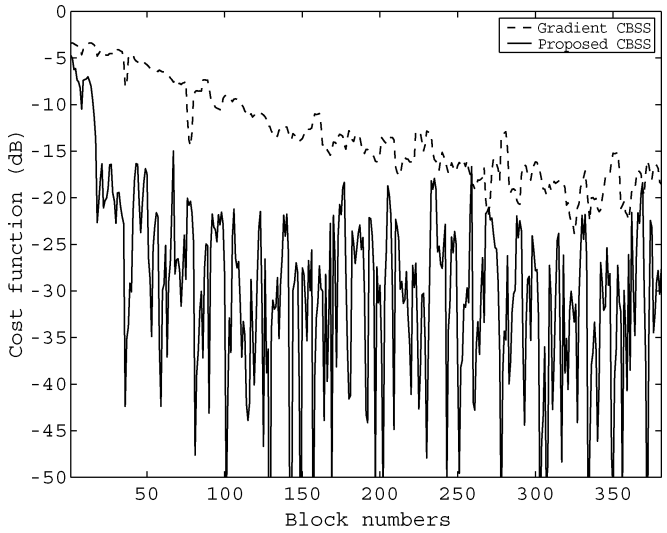

Fig. 15. Learning curves for the proposed recursive CBSS algorithm and the gradient-based CBSS algorithm: experimental results.

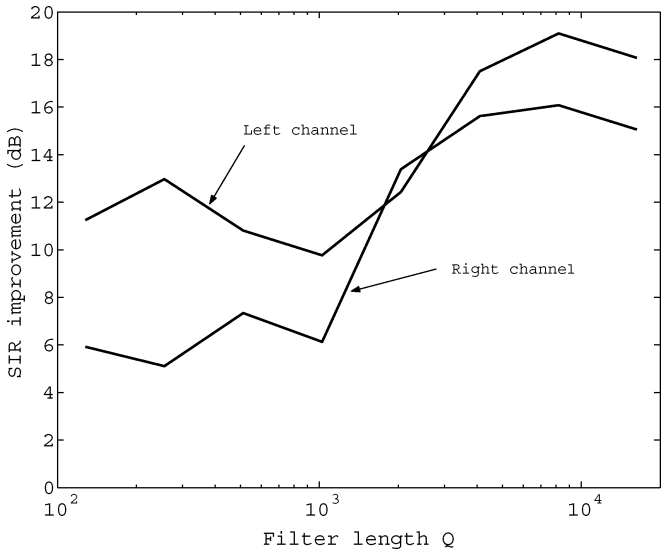

Fig. 16. Separation performance for two speakers recorded with two microphones.

sentences aloud in this environment; two microphones spaced 10.0 cm apart were used to input this audio. The input signals were digitized to 16 bits per sample, with a sample rate of 44.1 kHz. The vehicle compartment ambience was corrupted by noise from the car engine and other directionless noise.

The learning curve presented in Fig. 15 also clearly shows the superior convergence rate of the proposed recursive CBSS over the conventional gradient-based CBSS algorithm.

For the analysis presented in Fig. 15, $\lambda = 0.95$ for the recursive algorithm and $\mu = 0.01$ for the gradient CBSS-optimized algorithm. The observation signals were real-world recordings acquired in the car environment shown in Fig. 14. The filter lengths in the FIR filter matrix were set to 2048 and the length T of the DTFT was 8192. The sample block size was 8192, and the number of samples shifting L between neighboring blocks was 2048. This means that 75% ($K = 6144\,\text{samples}$) of the samples overlapped in two consecutive blocks. Compared to the simulation cases presented in the last subsection, in this experimental case we employed a longer sample block. We did this because (a) the sampling rate was higher, and (b) many more reflection paths were present in the real-world recording environment.

Convergence occurred in 70 iterations. Because one processing iteration was required for each block, the convergence time was $70 \times 2048/44\,100 = 3.25$ s.

For evaluation of separation performance in realistic environments, we report the improvement in terms of the signal to interference ratio (SIR) (e.g., [20], [33]–[35]), which is defined as

$$\text{SIR}_i = \frac{\hat{\sigma}_{iT}^2}{\hat{\sigma}_{iI}^2} \tag{37}$$

for output i, where $\hat{\sigma}_{iT}^2$ is the estimated power of target component $\sum_\tau g_{ii}(\tau)s_i(t-\tau)$ and $\hat{\sigma}_{iI}^2$ is the estimated power of interference components $\sum_{k\neq i}\sum_\tau g_{ik}(\tau)s_k(t-\tau)$ in the output. Fig. 16 shows the results for varying filter lengths on the separation of two speaker mixtures recorded with two microphones. In all experiments we used $T/L = 4$. The improvement in SIR can be as high as 19.0 dB for recordings obtained in our environment. As expected, the performance initially increases with increasing filter length, as the room can be modeled more accurately. However, larger filters require more separation parameters, which makes it more difficult to estimate them, thus the performance eventually degrades.

We also compare the performance of the proposed algorithm with that of several typical conventional algorithms, the multichannel blind deconvolution (MBD) [14], the convolutive blind separation of nonstationary sources (CBSS-NS) [20], the nonholonomic convolutive BSS (NH-CBSS) [36], the linear-prediction-based convolutive BSS (LP-CBSS) [33] and the natural gradient convolutive BSS (NG-CBSS) [33]. Table I lists the SIR's for the original and separated signals for two different two-channel signal separation benchmarks, so-called "Lee number" (rss_mA and rss_mB) and "Lee news" (rssd_A and rssd_B) [2]. "Lee number" was recorded in a typical office room. The distance between the speakers and the microphones was about 60 cm, arranged in a square. "Lee news" was recorded in a conference room (5.5 m $\times$ 8.0 m) in the presence of some air-conditioning noise. The distance between the speakers and the microphones was set to 120 cm to make the recordings noisier. Because of the longer impulse response and the louder noise, "Lee news" was more difficult to separate successfully than was "Lee number." Both benchmarks were recorded at a sampling rate of 16.0 kHz. We used these benchmarks because they have been used previously in numerous studies related to CBSS. The performance on these benchmarks can be used to compare different algorithms. Comparing these results, we see that our proposed algorithm can provide a better overall SIR performance. Although, in the case of "Lee news," the SIR performance for the left channel is a little worse than that of NG-MBD and LP-CBSS, the proposed algorithm provides a much better SIR performance for the right channel. From Table I, we can see that the conventional algorithms produced separation performances with large differences for the left and right channels. However, the proposed algorithm can produce a much better averaged separation performance for all of channels.

TABLE I
COMPARING SIR PERFORMANCE WITH OTHER ALGORITHMS

	Observation: Lee Number (rss_mA, rss_mB)		Observation: Lee News (rssd_A, rssc_B)	
	SIR(Left)	SIR(Right)	SIR(Left)	SIR(Right)
Observation	0.5 dB	0.9 dB	5.1 dB	0.9 dB
NG-MBD	15.1 dB	10.5 dB	17.9 dB	1.8 dB
CBSS-NS	15.3 dB	14.7 dB	14.8 dB	12.7 dB
NH-CBSS	16.9 dB	3.1 dB	14.1 dB	6.5 dB
LP-CBSS	20.3 dB	9.4 dB	17.8 dB	9.0 dB
NG-CBSS	21.0 dB	6.0 dB	12.8 dB	10.9 dB
Proposed algorithm	23.7 dB	18.6 dB	16.5 dB	15.6 dB

V. DISCUSSION

A. Overlearning Problem

As mentioned in Section III, our normal equation is an approximate one that describes the independence between sources. In most realistic situations, some extent of cross correlation exists between any two independently produced audio sources. In other words, the two sources do not yield a zero value for the cost function. If we force the outputs to have a zero cost value, the so-called overlearning problem arises. With conventional CBSS approaches, the cost function only needs to be minimal, but not necessarily zero. Thus, the overlearning problem can be avoided to some extent. However, if the outputs satisfy the normal equation for CBSS, the cost function (16) for the outputs vanishes. One might be concerned about whether the overlearning problem exists and how it affects our separation results.

As shown in Fig. 3, there is indeed, to some extent, overlearning in our separation criterion. The cost function does not become exactly zero, even for two independently generated sources. However, when checking the value of the cost function for any independently generated audio signals, we found that it is very negligible or is small. For example, for the two audio signals shown in Fig. 1, the cost-function value is as small as -30.10 dB. In other experiments, we found that such a value is approximately correct for most independently generated audio sources. The floors of the learning curves of our algorithm did not converge to zero but to some finite value. In the examples presented in Figs. 12, 13, and 15, the learning curves really converge to approximately -30.10 dB, although there were some fluctuations around these values. This fact indirectly proves the effectiveness of our algorithm, since there is not an obvious overlearning effect as expected.

B. Convergence of Learning Curves

One might wonder why convergence phenomena still exist when the algorithm is claimed to be explicit. This is because the estimated correlation matrix is less accurate at the beginning of processing and becomes progressively more accurate as more signal samples are processed over time. In other words, although one can always solve the normal equation exactly, its solution will still not correspond to the true optimum point if the correlation matrix in the normal equation is not perfectly estimated. Therefore, the convergence time of our algorithm is simply the elapsed time for exactly estimating the correlation matrix.

If the mixing system is static, a monotonically decreasing convergence curve is expected. If the transporting environment varies, some fluctuations in the convergence curve will exist. If the algorithm has momentarily converged to some extent and then the environment changes momentarily, the cost function increases at the moment corresponding to a fluctuation. The estimated value of the correlation matrix will be updated from this point. As time passes, the estimate will again become exact, and the cost function will converge again. In fact, the many fluctuations in Figs. 12, 13, and 15 are due to the frequent small environmental variances that occur during the audio recording. This includes subtle position changes of the sources (e.g., direction change of the speaker's face), and the effect of other extra sources transiently appearing and/or disappearing during the recording of the voices and the processing. Nonstationarity of the source also affects estimated results and is another reason for fluctuations.

Since our algorithm can converge within several seconds, it is applicable in cases in which the environment and speaker positions do not change rapidly, having stable time sessions as long as several seconds.

VI. CONCLUSION

In this paper, we proposed and investigated an approach for the real-time signal processing of CBSS, focusing on acoustic CBSS. We devised an overlap-and-save strategy that is more suitable for real-time CBSS processing. We introduced a normal equation for CBSS that provides a relationship between the separating matrix and correlation matrix of input signals. This normal equation corresponds to the minimum point of the separation criterion and it is based only on the second order statistics of signals in the frequency domain. We presented a real-time CBSS algorithm by solving the normal equation and estimating the correlation matrix.

One important feature of the algorithm is that it can *recursively* estimate the correlation matrix. Another favorable feature is that it can *explicitly*, instead of stochastically (as occurs in conventional approaches), solve the normal equation for CBSS to find the separating matrix.

The relationship between this approach and the conventional gradient-based approach is analogous to the relationship between the RLS approach and the LMS approach in adaptive filtering with supervised learning.

Our algorithm realizes real-time separation of the convolutive mixtures of acoustic sources. In simulations, the algorithm has a superior convergence rate over its counterpart, the gradient-based approach. In the two presented examples, the convergence times are about 0.64 and 1.28 s.

We have also applied the algorithm to real-time CBSS processing in realistic environments to separate acoustic sources. Acoustic signals were separated by processing signal samples block by block. In one instance, the sampling rate of input signals was 44.1 kHz, and each block contained 8192 signal samples. These experimental results demonstrated that our algorithm can converge to a stable separating state within several seconds, which is much faster than the convergence time of the gradient-based algorithm. The convergence time for our proposed algorithm in the real-world test situation is about 3.25 s.

At present, we have only realized a recursive CBSS algorithm with a cost function based on second-order signal statistics. This approach can be also applied to other cost functions based on

higher-order signal statistics. We hope to report such studies in the future.

APPENDIX
DERIVATION OF THE RELATION BETWEEN THE MODIFIED CORRELATION MATRIX AND THE TIME-LAGGED CROSS CORRELATIONS

From (5), we obtain the following decomposition

$$\begin{aligned}\boldsymbol{R}_y(\omega_b,n) &= \sum_{k=1}^{n}\beta(n,k)\phi\left(\boldsymbol{y}(\omega_b,k)\right)\phi\left(\boldsymbol{y}^H(\omega_b,k)\right)\\ &= \beta(n,1)\bar{\boldsymbol{y}}(\omega_b,1)\bar{\boldsymbol{y}}^H(\omega_b,1)+\ldots\\ &\quad+\sum_{l=1}^{D-1}\sum_{m=1}^{D-1}\beta(n,D-1)\bar{\boldsymbol{y}}(\omega_b,l)\bar{\boldsymbol{y}}^H(\omega_b,m)\\ &\quad+\sum_{k=D}^{n}\sum_{l=k-D+1}^{k}\sum_{m=k-D+1}^{k}\beta(n,k)\\ &\quad\cdot\bar{\boldsymbol{y}}(\omega_b,l)\bar{\boldsymbol{y}}^H(\omega_b,m) \qquad \text{(A.38)}\end{aligned}$$

where $\bar{\boldsymbol{y}}(\omega_b,l)$ is T-length discrete-time Fourier transform of L-length signal block, i.e., $\bar{\boldsymbol{y}}(\omega_b,l) = \mathrm{DTFT}([0_1,\ldots,0_{L(l-1)},\boldsymbol{y}(L(l-1)+1),\ldots,\boldsymbol{y}(L(l-1)+L)])$. If $\beta(n,k)$ is used as defined by (6), and when n is sufficiently large, then $\boldsymbol{R}_y(\omega_b,n)$ can be approximated as follows:

$$\boldsymbol{R}_y(\omega_b,n)\approx\sum_{k=D}^{n}\sum_{l=k-D+1}^{k}\sum_{m=k-D+1}^{k}\beta(n,k)\bar{\boldsymbol{y}}(\omega_b,l)\bar{\boldsymbol{y}}^H(\omega_b,m). \qquad \text{(A.39)}$$

In (A.38) or (A.39), the right-hand side (RHS) includes both the $l=m$ terms and $l\neq m$ terms. That is, the modified correlation matrix (A.38) or (A.39) includes both the zero-lag correlation and the time-lagged correlations. The dependence on the time-lagged correlations is due to the overlap-and-save.

In the same way, $\boldsymbol{R}_x(\omega_b,n)$ also includes both the zero-lag correlation and the time-lagged correlations.

We can use a simple sequence processing method to decorrelate the correlation matrix $\boldsymbol{R}_y(\omega_b,n)$.

If we set all of the sample equal to zero, except for the last L samples, $\boldsymbol{R}_y(\omega_b,n)$ in (A.38) degenerates to

$$\boldsymbol{R}_y^{(0)}(\omega_b,n)\approx\sum_{k=D}^{n}\beta(n,k)\bar{\boldsymbol{y}}(\omega_b,k)\bar{\boldsymbol{y}}^H(\omega_b,k) \qquad \text{(A.40)}$$

i.e., there are only the zero-lag correlations.

If we extend the nonzero sample to the last $2L$ samples and suppose the zero-lag correlation in (A.40) has been decorrelated, we obtain

$$\boldsymbol{R}_y^{(1)}(\omega_b,n)\approx\sum_{k=D}^{n}\beta(n,k)\mathrm{Re}\left(\bar{\boldsymbol{y}}(\omega_b,k)\bar{\boldsymbol{y}}^H(\omega_b,k-1)\right). \qquad \text{(A.41)}$$

That is, only the 1-lagged correlation exists. In such a way, if we decorrelate the $0-,1-,\ldots,(l-1)$-lagged correlations for a positive integer $l<D$, and extend the nonzero sample to the last lL samples, only the l-lagged correlation exists.

$$\boldsymbol{R}_y^{(l)}(\omega_b,n)\approx\sum_{k=D}^{n}\beta(n,k)\mathrm{Re}\left(\bar{\boldsymbol{y}}(\omega_b,k)\bar{\boldsymbol{y}}^H(\omega_b,k-l)\right). \qquad \text{(A.42)}$$

The 0-lagged correlation is just the equal time correlation.

ACKNOWLEDGMENT

The authors acknowledge the insightful comments provided by the anonymous reviewers which have added much to the clarity of this paper.

REFERENCES

[1] K. Torkkola, "Blind separation of convolved sources based on information maximization," in *Proc. IEEE Workshop on Neural Networks and Signal Processing (NNSP'96)*, Kyoto, Japan, 1996, pp. 423–432.
[2] T.-W. Lee, *Independent Component Analysis, Theory and Applications*. Boston, MA: Kluwer Academic, 1998.
[3] L. Parra, C. Spence, and B. D. Vries, "Convolutive blind source separation based on multiple decorrelation," in *Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP'98)*, Cambridge, U.K., 1998, pp. 23–32.
[4] L. Parra and C. Alvino, "Geometric source separation: merging convolutive source separation with geometric beamforming," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 6, pp. 352–362, Sep. 2002.
[5] S. Araki, S. Makino, R. Mukai, and H. Saruwatari, "Equivalence between frequency domain blind source separation and frequency domain adaptive null beamformers," in *Proc. Eurospeech2001*, 2001, pp. 2595–2598.
[6] J. Anemuller, T. Sejnowski, and S. Makeig, "Complex independent component analysis of frequency-domain eeg data," *Neural Netw.*, pp. 1311–1323, 2003.
[7] W. Wang, J. A. Chambers, and S. Sanei, "A joint diagonalization method for convolutive blind separation of nonstationary sources in the frequency domain," in *Proc. ICA2003*, 2003, pp. 939–944.
[8] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, pp. 1545–1553, 2002.
[9] M. Joho and H. Mathis, "Joint diagonalization of correlation matrices by using gradient methods with application to blind signal separation," in *Proc. SAM*, 2002, pp. 273–277.
[10] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE Trans. Signal Process.*, vol. 49, pp. 530–541, 2001.
[11] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1–4, pp. 1–24, Aug. 2001.
[12] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks*, vol. 8, no. 3, pp. 411–419, 1995.
[13] M. Kawamoto, K. Matsuoka, and M. Ohnishi, "A method for blind separation for convolved nonstationary signals," *Neurocomputing*, vol. 22, pp. 157–171, 1998.
[14] R. H. Lambert and C. L. Nikias, "Blind deconvolution of multipath mixtures," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed: Wiley, 2000, vol. I, pp. 377–436.
[15] J. V. de Laar, E. Habets, J. Peters, and P. Lokkart, "Adaptive blind audio signal separation on a DSP," in *Proc. ProRISC 2001*, 2002, pp. 475–479.
[16] S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," in *Proc. ICA2001*, 2001, pp. 651–656.
[17] S. Ding, M. Otsuka, M. Ashizawa, T. Niitsuma, and K. Sugai, "Blind source separation of real-world acoustic signals based on ICA in time-frequency domain," IEICE, Tech. Rep. IEICE, pp. 1–8, vol. EA2001-1, 2001.
[18] S. Ding, T. Hikichi, T. Niitsuma, M. Hamatsu, and K. Sugai, "Recursive method for blind source separation and its applications to real-time separations of acoustic signals," in *Proc. ICA2003*, 2003, pp. 517–522.
[19] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York: Wiley, 2003.
[20] L. Parra and C. Spence, "Convolutive blind separation of nonstationary sources," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 320–327, Mar. 2000.
[21] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs: Prentice-Hall, 1996.
[22] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Phys. Rev. Lett.*, vol. 72, no. 23, 1994.
[23] A. Ziehe and K.-R. Müller, "TDSEP—An efficient algorithm for blind separation using time structure," in *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, Skvde, Sweden, 1998, pp. 675–680.
[24] M. Ikram and D. Morgan, "Exploring permutation inconsistency in blind separation of signals in a reverberant environment," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2000, pp. 1041–1044.
[25] A. Cichocki, R. Umbehauen, and E. Rummert, "Robust learning for blind separation of signals," *Electron. Lett.*, vol. 30, no. 17, pp. 1386–1387, Aug. 1994.

[26] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.
[27] S. Fiori, "Fully-multiplicative orthogonal-group ica neural algorithm," *Electron. Lett.*, vol. 39, no. 24, pp. 1737–1738, Nov. 2003.
[28] J. K. Tugnait, "Channel estimation and equalization using high-order statistics," in *Signal Processing Advances in Wireless and Mobile Communications*, G. Giannakis, Ed. Upper Saddle River, NJ: Prentice-Hall, 2000, vol. I, pp. 1–40.
[29] Y. Inouye and S. Ohno, "Adaptive algorithms for implementing the single-stage criterion for multichannel blind deconvolution," in *Proc. 5th Int. Conf. on Neural Information Processing (ICONIP'98)*, 1998, pp. 733–736.
[30] L.-Q. Zhang, A. Cichocki, and S. Amari, "Multichannel blind deconvolution of nonminimum phase systems using filter decomposition," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1430–1441, 2004.
[31] Blind Source Separation: Audio Examples, T.-W. Lee. (2002). [Online]. Available: http://inc2.ucsd.edu/~tewon/
[32] ICA '99 Speech Signals, D. W. E. Schobben. (1999). [Online]. Available: http://www2.ele.tue.nl/ica99//realworld2.html
[33] S. C. Douglas and X. Sun, "Convolutive blind separation of speech mixtures using the natural gradient," *Speech Commun.*, vol. 39, pp. 65–78, 2003.
[34] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Convolutive blind source separation for more than two sources in the frequency domain," in *ICASSP2004*, vol. III, 2004, pp. 885–888.
[35] D. Schobben, K. Torkkola, and P. Smaragdis, "Evaluation of blind signal separation methods," in *Proc. Int. Workshop Independent Component Analysis and Blind Signal Separation*, 1999, pp. 261–266.
[36] S. C. Douglas, "Blind signal separation and blind deconvolution," in *Handbook of Neural Networks Signal Processing*, Y.-H. Hu and J.-N. Hwang, Eds. Boca Raton, FL: CRC, 2001, ch. 7.

Shuxue Ding (M'04) received the M.Sc. degree in physics from the Dalian University of Technology, P. R. China, in 1988, and the Ph.D. degree in physics from Tokyo Institute of Technology, Japan, in 1996.

From 1989 to 1991 and from 1991 to 1992, respectively, he was an Assistant Professor and Associate Professor with the Dalian University of Technology. From 1996 to 1998, he was with Fujisoft-ABC Inc., Japan, where he was involved in algorithm design for telecommunication systems. From 1998 to 2003, he was with Clarion Company, Ltd, Japan, where he engaged in research in communication and signal processing, especially in speech recognition. From 2003 to 2005, he was a visiting faculty member with the University of Aizu, Japan. He is currently an Associate Professor with the Department of Computer Software, University of Aizu, and an Associate Research Scientist of the Brian Science Institute, Institute of Physical and Chemical Research (RIKEN), Japan. He has been engaged in research in a wide range of areas of mathematical and physical engineering, such as statistical signal processing, optimization, neural computation, bioelectromagnetism, and information sciences. In particular, he has devoted himself to blind source separation (BSS) and independent component analysis, and their applications in acoustic signals and vital signs. Recently, he is also conducting research in brain-style information processing, pattern recognition, and BSS, from the viewpoint of generalized information based on algorithmic, i.e., Kolmogorov, complexity. He is also interested in speech and image processing, quantum computation, quantum and information, and the physics of information.

Dr. Ding is a member of IEICE, Japan.

Jie Huang (M'94) received the B.Eng. degree from the Nagoya Institute of Technology, Japan, in 1985, and the M.Eng. and D.Eng. degrees from Nagoya University, Japan, in 1987 and 1991, respectively.

From 1992 to 1993, he was a system engineer with Phenix Data Corp. He worked as a frontier research scientist with the Bio-Mimetic Control Research Center under the Institute of Physical and Chemical Research (RIKEN), Japan, from 1993 to 1998. He was an Assistant Professor with the School of Computer Science and Engineering, University of Aizu, from 1998 to 2002. Since 2002, he has been an Associate Professor with the University of Aizu. His research interests include digital signal processing, human audition, and robot sensing systems.

Dr. Huang received the Technical Development Award from SICE in 1996. He is a member of RSJ, SICE, IEICE, ASJ, ans ASA.

Daming Wei (M'92) received the B.Eng. degree from the Department of Mathematics and Mechanics, Tsinghua University, Beijing, China. He received the M.Eng. degree in computer engineering from the Shanghai Institute of Technology (Shanghai University), and the Ph.D. degree in biomedical engineering from Zhejiang University, China.

He was a Deputy Director of the Biomedical Engineering Section, Zhejiang University, before he joined the Tokyo Institute of Technology, Japan, in 1986. Since then, he has been with several companies and universities in Japan. He is currently a Professor and serves as Director of the Department of Computer Software, University of Aizu, Japan. He has long-term experiences and well-known achievements in developing the state-of-the-art computer heart model and simulation of electrocardiogram. Recent directions in his research group include biomedical modeling, computer simulation and visualization, signal processing, and e-health. He has authored a large number of journal and conference papers, several books, and book chapters. He is an inventor of a number of U.S. and Japanese patents. He is currently leading several large-scale research projects funded by central and domestic government aimed at academia-industry collaboration.

Prof. Wei is a member of the New York Academy of Sciences. He serves as a council member of the International Society of Bioelectromagnetism and an Associate Editor of the *International Journal of Bioelectromegnetism*. He is a council member of Japan Biomedical Engineering Society Tohoku Branch.

Andrzej Cichocki (M'96) received the M.Sc. (with honors), Ph.D., and Habilitate Doctorate (Dr.Sc.) degrees, all in electrical engineering, from Warsaw University of Technology, Poland, in 1972, 1975, and 1982, respectively.

Since 1972, he has been with the Institute of Theory of Electrical Engineering and Electrical Measurements, Warsaw University of Technology, where he became a full Professor in 1991. He was with the University Erlangen-Nuernberg (Germany) for a few years as an Alexander Humboldt Research Fellow and Guest Professor working in the area of artificial neural networks. He conducted and realized several research projects there which have obtained very good evaluation by experts. During 1996–1999, he was working as a Team Leader of the Laboratory for Artificial Brain Systems at the Frontier Research Program RIKEN (Japan), in the Brain Information Processing Group. Currently, he is Head of the Laboratory for Advanced Brian Signal Processing, Brain Science Institute RIKEN, Group Brain-like Information Systems, directed by Prof. Amari. His current research interests include signal and image processing (especially blind signal/image processing), neural networks and their applications, learning theory and algorithms, generalization and extensions of independent and principal component analysis, optimization problems, nonlinear circuits and systems theory and their applications, and artificial intelligence. He is the coauthor of three international and successful books: *Adaptive Blind Signal and Image Processing* (New York: Wiley, 2002), *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems* (New York: Springer-Verlag, 1989), and *Neural Networks for Optimization and Signal Processing* (New York: Wiley and Teubner Verlag, 1993/1994). He is also an author or coauthor of more than 150 papers.

Dr. Cichocki is a member of several international scientific committees and the Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS (since January 1998). He is a reviewer of several international journals, e.g., the IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON BIOLOGICAL CYBERNETICS, IEEE TRANSACTIONS ON ELECTRONICS LETTERS, NEUROCOMPUTING, NEURAL COMPUTATION, and IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING. He is a member of the IEEE Signal Processing and Neural Networks Societies.